

Учреждение образования  
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**Кафедра информационных систем и технологий**

# **ОСНОВЫ ДИСКРЕТНОЙ МАТЕМАТИКИ И ТЕОРИИ АЛГОРИТМОВ**

**Программа, методические указания  
к выполнению лабораторных и контрольных работ  
по дисциплине «Основы дискретной математики  
и теории алгоритмов» для студентов  
специальности 1-40 01 02-03 «Информационные системы  
и технологии (издательско-полиграфический комплекс)»**

Минск 2014

УДК 519.1(075.8)  
ББК 22.17я73  
О75

Рассмотрены и рекомендованы редакционно-издательским советом Белорусского государственного технологического университета

Составитель *А. П. Лащенко*

Рецензент  
доктор физико-математических наук, профессор,  
заведующий кафедрой высшей математики БГТУ  
*В. М. Марченко*

По тематическому плану учебно-методической литературы на 2013 год. Поз. 168.

Для студентов специальности 1-40 01 02-03 «Информационные системы и технологии (издательско-полиграфический комплекс)» по дисциплине «Основы дискретной математики и теории алгоритмов».

УДК 519.1(075.8)  
ББК 22.17я73

© УО «Белорусский государственный  
технологический университет», 2014

## ВВЕДЕНИЕ

Задачами преподавания прикладных математических дисциплин являются: выяснение сущности научного подхода к описанию и исследованию реальных производственных и социальных процессов, роли математических методов в системе естественнонаучных дисциплин как способе познания окружающего мира; развитие у обучаемых способности к логическому и алгоритмическому мышлению, умения, знаний и приемов исследования и решения математически формализованных задач.

На нынешнем этапе развития инженерно-технического образования в области современных информационных технологий дискретная математика предстает как язык общения «цивилизованных» инженеров.

Современный специалист обязан владеть основами математического моделирования и его реализации в компьютерных информационных технологиях, чтобы быть конкурентоспособным и выдерживать темпы научно-технического прогресса. Математические методы дискретной математики выступают в этой связи как возможность дать унифицированный научный подход к изучению различных физических и социальных явлений реального мира путем составления их математических моделей, которые во многих случаях описываются одними и теми же математическими структурами. Таким образом, математическое моделирование на основе дискретной математики и теории алгоритмов позволяет не только изучить общие закономерности различных производственных задач, но и дать универсальные рекомендации по их решению.

Распределение часов по курсу «Основы дискретной математики и теории алгоритмов» для студентов специальности «Информационные системы и технологии (издательско-полиграфический комплекс)» дано в следующей таблице.

Форма обучения	Количество часов по учебному плану			Форма проверки знаний
	Установочная сессия	Экзаменационная сессия		
	Лекции	Лекции	Лабораторные	
Заочная	8	4	6	зачет
Дневная	—	34	34	экзамен

## **1. ОБЩИЕ УКАЗАНИЯ**

Основной формой обучения студента является самостоятельная работа. Рекомендуемые учебные пособия по данному курсу для специальности 1-40 01 02-03 «Информационные системы и технологии (издательско-полиграфический комплекс)» приведены в списке литературы. Если в процессе изучения материала или при выполнении контрольных работ у студента возникают трудности, то следует обратиться на кафедру «Информационные системы и технологии» университета (тел. 327-43-76) для получения устной или письменной консультации.

## 2. ПРОГРАММА КУРСА «ОСНОВЫ ДИСКРЕТНОЙ МАТЕМАТИКИ И ТЕОРИИ АЛГОРИТМОВ»

Элементы и множества. Способы задания множеств. Представление множеств на компьютере. Алгебра подмножеств. Операции над множествами. Декартово произведение множеств. Бинарные отношения. Свойства бинарных отношений. Представление отношений на компьютере. Функции. Функциональные отношения. Инъекция, сюръекция и биекция. Суперпозиция функций. Представление отношений на компьютере. Упорядоченные множества.

Элементарные булевы функции. Способы задания логических функций. Булевы функции одной переменной. Булевы функции двух переменных. Алгебра булевых функций. Нормальные формы логических функций. Полнота и замкнутость. Минимизация логических функций. Визуально матричный метод. Применение логических функций. Представление булевых функций в компьютере. Табличные представления. Строковые представления. Основные понятия логики предикатов. Применение выражений логики предикатов для описания некоторых отношений.

Перечислительные и оптимизационные комбинаторные задачи. Комбинаторные конфигурации: сочетания, перестановки и размещения. Графическое представление перестановок. Методы комбинаторного поиска. Производящие функции. Дерево поиска. Принцип включения и исключения.

Основные понятия и определения графов. Орграфы, псевдографы, мультиграфы и гиперграфы. Маршруты, цепи, циклы. Виды графов и операции над графами. Представление графов на компьютере. Матрицы смежности и инцидентности. Деревья. Построение остовых деревьев. Независимые и доминирующие множества. Раскраска и планарность графов. Паросочетания в графе. Кратчайшие пути и алгоритмы их поиска. Задача о покрытии булевой матрицы и родственные с ней оптимизационные задачи на графах.

## 3. ЭЛЕМЕНТЫ ТЕОРИИ МНОЖЕСТВ И ОТНОШЕНИЙ

### 3.1. Множества. Основные понятия и определения

Понятия «множества», «отношения», «функции» и близкие к ним составляют основной словарь дискретной (равно как и «непрерывной») математики.

Человеческое мышление устроено так, что мир представляется состоящим из отдельных «объектов», хотя ясно, что мир – единое неразрывное целое, и выделение в нем объектов – это не более чем произвольный акт нашего сознания, позволяющий сформировать доступную для рационального анализа картину мира. Но как бы то ни было, выделение объектов и их совокупностей – естественный (или даже единственно возможный) способ организации нашего мышления, поэтому неудивительно, что он лежит в основе главного инструмента описания точного знания – математики.

Понятие множества принадлежит к числу фундаментальных неопределяемых понятий математики. Можно сказать, что множество – это любая определенная совокупность объектов. Объекты, из которых составлено множество, называются его элементами. Интуитивно под множеством будем понимать совокупность определенных вполне различаемых объектов, рассматриваемых как единое целое.

Можно говорить о множестве стульев в комнате, людей, живущих в городе, студентов в группе, о множестве натуральных чисел, букв в алфавите, состояний системы и т. п. При этом о множестве можно вести речь только тогда, когда отдельные объекты, его составляющие, различимы между собой. Например, нельзя говорить о множестве капель в стакане воды, так как невозможно четко и ясно указать каждую отдельную каплю. Обычно множества обозначают прописными буквами латинского алфавита, а элементы множеств – строчными буквами.

*Пример 1.* Множество  $S$  страниц в данной книге. Множество  $N$  натуральных чисел: 1, 2, 3, ... . Множество  $P$  простых чисел: 2, 3, 5, ... . Множество  $Z$  целых чисел: ... , -2, -1, 0, 1, 2, ... . Множество  $R$  вещественных чисел. Множество  $A$  различных символов на этой странице.

Желательно, чтобы элементы множеств были четко описанными и неизменяемыми объектами.

Принадлежность элемента  $a$  множеству  $M$  обозначается  $a \in M$  (« $a$  принадлежит  $M$ »); не принадлежность  $a$  множеству  $M$  обозначается  $a \notin M$  (« $a$  не принадлежит  $M$ »). Записью  $a_1, a_2, \dots, a_n \in M$  пользуются в качестве сокращения для записи  $a_1 \in M, a_2 \in M, \dots, a_n \in M$ .

Обычно в конкретных рассуждениях элементы всех множеств берутся из некоторого одного, достаточно широкого множества  $U$  (своего для каждого случая), которое называется универсальным множеством (или универсумом).

Важным понятием теории множеств является понятие пустого множества. Пустым называется множество, не содержащее ни одного элемента. Оно обозначается  $\emptyset$ . Пустое множество введено в математике для удобства и единообразия языка. Например, если исследуется множество объектов, обладающих каким-либо свойством, и впоследствии выясняется, что таких объектов не существует, то гораздо удобнее назвать исследуемое множество пустым, чем несуществующим. Утверждение «множество  $M$  непустое» является более компактной формулировкой равносильного ему утверждения «существуют элементы, принадлежащие  $M$ ».

Множество  $A$  называется подмножеством множества  $B$  (обозначение  $A \subseteq B$ ; знак  $\subseteq$  называется знаком включения), если любой элемент множества  $A$  принадлежит множеству  $B$ . При этом говорят, что  $B$  содержит или покрывает  $A$ . Например, пусть  $A$  – множество всех отличников группы,  $B$  – множество всех студентов группы. Тогда  $A \subseteq B$ . Принято считать, что пустое множество является подмножеством любого множества. Множества  $\emptyset$  и  $A$  называются несобственными подмножествами множества  $A$ . Все остальные подмножества называются собственными подмножествами множества  $A$ .

Множества  $A$  и  $B$  равны ( $A = B$ ), если они состоят из одних и тех же элементов, иначе говоря, если  $A \subseteq B$  и  $B \subseteq A$ . Второй вариант определения равенства множеств указывает на наиболее типичный метод доказательства того, что данные множества равны, заключающийся в доказательстве сначала утверждения  $A \subseteq B$ , а затем  $B \subseteq A$ . Из определения равенства множеств вытекает, что порядок элементов во множестве несуществен. Множества  $A$  и  $B$  не равны (обозначение  $A \neq B$ ), если либо во множестве  $A$  есть элементы, не принадлежащие  $B$ , либо во множестве  $B$  есть элементы, не принадлежащие  $A$ .

Если  $A \subseteq B$  и  $A \neq B$ , то  $A$  часто называют собственным, строгим или истинным подмножеством  $B$  (обозначение  $A \subset B$ ; знак  $\subset$  называется знаком строгого включения). Если желают подчеркнуть, что

множество  $B$  содержит и другие элементы кроме элементов из множества  $A$ , то используют символ строгого включения. Например, множество целых чисел  $Z$  является собственным подмножеством множества рациональных чисел  $Q$ ; таким образом,  $Z \subset Q$ .

Множества бывают конечными и бесконечными. Множество называют конечным, если число его элементов конечно, т. е. если существует натуральное число  $n$ , равное количеству элементов множества. Множество называют бесконечным, если оно содержит бесконечное число элементов. Пустое множество будем условно относить к конечным множествам.

*Задание множеств.* Чтобы задать множество, нужно указать, какие элементы ему принадлежат. Это можно сделать различными способами:

- перечислением элементов:  $M := \{a_1, a_2, \dots, a_k\}$ ;
- характеристическим предикатом:  $M := \{x \mid P(x)\}$ ;
- порождающей процедурой:  $M := \{x \mid x := f\}$ .

При задании множеств перечислением обозначения элементов обычно заключают в фигурные скобки и разделяют запятыми. Характеристический предикат – это некоторое условие, выраженное в форме логического утверждения или процедуры, возвращающей логическое значение, и позволяющее проверить, принадлежит ли любой данный элемент множеству. Если для данного элемента условие выполнено, то он принадлежит определяемому множеству, в противном случае – не принадлежит. Порождающая процедура – это процедура, которая в процессе работы порождает некоторые объекты, являющиеся элементами определяемого множества.

*Пример 2.*  $M_9 := \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ ;  $M_9 := \{n \mid n \in N \ \& \ n < 10\}$ ;  $M_9 := \{n \mid n := 0; \text{ for } i \text{ from } 1 \text{ to } 9 \text{ do } n := n + 1; \text{ yield } n \text{ end for}\}$ .

Перечислением можно задавать только конечные множества. Бесконечные множества задаются характеристическим предикатом или порождающей процедурой.

Из определения множества следует, что в нем не должно быть неразличимых элементов. Поэтому во множестве не может быть одинаковых элементов. Запись множества  $\{2, 2, 3, 5\}$  следует рассматривать как некорректную и заменить ее на  $\{2, 3, 5\}$ . Так, множество всех простых делителей числа 60 равно  $\{2, 3, 5\}$ .

*Пример 3.* Даны два множества:  $A = \{0, 1\}$  и  $B = \{\{0, 1\}\}$ .

$A$  является элементом  $B$ , поэтому справедлива запись  $A \in B$ , но не  $A \subseteq B$ .  $A \neq B$ , так как элементами  $A$  являются 0 и 1, а единственным



элементом  $B$  является множество  $A$ , элементы множеств  $A$  и  $B$  имеют различную природу.

*Пример 4.* Даны три множества:  $A = \{0, 1\}$ ,  $B = \{\{0, 1\}, 2\}$ ,  $C = \{\{\{0, 1\}, 2\}, 3\}$ .  $A \in B$ , но  $A \notin C$ , а  $B \in C$ .  $A$  – элемент, но не подмножество  $B$ ;  $B$  – элемент, но не подмножество  $C$ . Элементами  $A$  являются 0 и 1, элементами  $B$  – множество  $A$  и 2, а элементами  $C$  являются множество  $B$  и 3. Поэтому  $A \notin C$ .

*Пример 5.* Пусть  $A = \{a_1, a_2, \dots, a_n\}$ ,  $n \in N$  – некоторое конечное множество. Каждому подмножеству  $B \subseteq A$  сопоставим двоичное слово длиной  $n$  по следующему правилу:

$$S_B = \alpha_1, \dots, \alpha_n, \text{ где } \alpha_i = \begin{cases} 1, & \text{если } a_i \in B, \\ 0, & \text{если } a_i \notin B, \end{cases} \quad i = \overline{1, n}.$$

Таким образом, каждому подмножеству множества  $A$  ставится в соответствие единственное двоичное слово длиной  $n$ , и каждому двоичному слову длиной  $n$  ставится в соответствие единственное подмножество множества  $A$ .

Для множества  $A = \{1, 2, 3, 4\}$ :

1) построить его подмножества по следующим двоичным словам: 0101, 1111, 1001, 1000, 0100, 0110, 0111, 0001;

2) найти двоичные слова, соответствующие его подмножествам:  $\{3\}$ ,  $\{4, 3\}$ ,  $\{1, 3\}$ ,  $\emptyset$ ,  $\{2, 3, 4\}$ ,  $\{1, 4, 3\}$ ,  $\{1, 2, 3\}$ ,  $\{1, 2, 3, 4\}$ .

1. Получаем, согласно вышеуказанному правилу, следующие множества:  $\{2, 4\}$ ,  $\{1, 2, 3, 4\} = A$ ,  $\{1, 4\}$ ,  $\{1\}$ ,  $\{2\}$ ,  $\{2, 3\}$ ,  $\{2, 3, 4\}$ ,  $\{4\}$ .

2. Получаем, согласно вышеуказанному правилу, следующие двоичные слова длиной 4: 0010, 0011, 1010, 0000, 0111, 1011, 1110, 1111.

*Равномощные множества.* Взаимно однозначным соответствием между двумя множествами  $A$  и  $B$  называется такое правило (или закон)  $f$ , по которому каждому элементу  $a \in A$  ставится в соответствие единственный элемент  $f(a) \in B$ , для любого элемента  $b \in B$  существует единственный элемент  $a \in A$ , такой, что  $f(a) = b$ .

Множества  $A$  и  $B$  называются равномощными (обозначение  $A \leftrightarrow B$ ), если между ними можно установить взаимно однозначное соответствие. В этом случае говорят также, что множества  $A$  и  $B$  изоморфны.

*Пример 6.* Соответствие  $n \mapsto 2n$  устанавливает взаимно однозначное соответствие между множеством натуральных чисел  $N$  и множеством четных натуральных чисел  $2N$ .

Нетрудно видеть, что:

1) любое множество взаимно однозначно соответствует самому себе:  $A \leftrightarrow A$  – достаточно рассмотреть соответствие  $a \mapsto a$ , где  $a \in A$ ;

2) если  $A \leftrightarrow B$ , то  $B \leftrightarrow A$  – достаточно использовать соответствие  $a \mapsto b$  для построения соответствия  $b \mapsto a$ ;

3) если множество  $A$  равномощно  $B$ , а  $B$  равномощно  $C$ , то  $A$  равномощно  $C$ , т. е.  $(A \leftrightarrow B) \& (B \leftrightarrow C) \Rightarrow (A \leftrightarrow C)$ .

*Пример 7.* Множество десятичных цифр равномощно множеству пальцев на руках человека, но не равномощно множеству пальцев на руках и на ногах.

*Пример 8.* Показать, что множества  $R^2$  и  $A \times B$ , где  $A = \{(x, y) \in R^2 \mid 2x + y = 1\}$ ,  $B = \{(x, y) \in R^2 \mid x - y = 0\}$  равномощны.

Нетрудно видеть, что  $R^2$  равномощно множеству всех точек на действительной плоскости,  $A$  равномощно множеству всех точек прямой  $2x + y = 1$ ,  $B$  – множеству всех точек прямой  $x - y = 0$  на действительной плоскости. Поскольку коэффициенты при  $x$  и  $y$  у данных двух прямых не пропорциональны, прямые пересекаются на плоскости в единственной точке с координатами  $(1/3, 1/3)$ . Для доказательства равномощности  $R^2$  и  $A \times B$  достаточно показать равномощность множества всех точек действительной плоскости и всех упорядоченных пар точек на прямых  $2x + y = 1$  и  $x - y = 0$ .

Каждой точке  $M$  на плоскости поставим в соответствие пару точек  $(A_M, B_M)$  на прямых  $2x + y = 1$  и  $x - y = 0$ , являющихся точками пересечения этих прямых с прямыми, проходящими через данную точку и параллельными  $x - y = 0$  и  $2x + y = 1$  соответственно. Если точка  $N$  принадлежит прямой  $2x + y = 1$  либо  $x - y = 0$ , то первым (вторым) элементом пары точек на прямых будет сама данная точка, а вторым (первым) элементом пары – точка  $(1/3, 1/3)$ .

*Добавление и удаление элементов.* Если  $A$  – множество, а  $x$  – элемент, причем  $x \notin A$ , то  $x$  можно *добавить* в  $A$ :

$$A + x = \{y \mid y \in A \vee y = x\}.$$

Аналогично, если  $A$  – множество, а  $x$  – элемент, причем  $x \in A$ , то  $x$  можно *удалить* в  $A$ :

$$A - x = \{y \mid y \in A \& y \neq x\}.$$

Легко увидеть, что при удалении и добавлении конечного числа элементов конечного множества остаются конечными, а бесконечные – бесконечными.

*Мощность конечного множества.* Число элементов в конечном множестве  $A$  называется мощностью  $A$  и часто обозначается  $|A|$ . Пустое множество, т. е. не содержащее элементов, является множеством мощности 0:  $|\emptyset| = 0$ .

*Теорема 1.* Любое непустое конечное множество равномощно некоторому отрезку натурального ряда:

$$\forall A \ A \neq \emptyset \ \& \ |A| < \infty \Rightarrow \exists k \in N \ |A| = |1..k|.$$

*Доказательство.*

Рассмотрим следующую программу.

```
i = 0
while A ≠ ∅ do
  select x ∈ A
  i := i+1
  x → i
  A := A – x
End while
```

*Следствие 1.* Любой отрезок натурального ряда конечен:  
 $\forall n \in N \ |1..n| < \infty$ .

*Следствие 2.* Различные отрезки натурального ряда неравномощны:  $n \neq m \Rightarrow |1..n| \neq |1..m|$ .

*Теорема 2.* Между конечными множествами  $A$  и  $B$  существует взаимно однозначное соответствие тогда и только тогда, когда их мощности равны  $|A| = |B|$ .

*Доказательство.* Пусть  $f: A \rightarrow B$  задает взаимно однозначное соответствие между множествами. Пусть  $|A| \neq |B|$ , допустим,  $|A| < |B|$ . Тогда, поскольку для  $\forall b \in B \ \exists a \in A$ , такое что  $f(a) = b$ , для некоторого  $b \in B$  найдутся несколько элементов из  $A$  с таким свойством, а это противоречит определению взаимно однозначного соответствия. Предположим теперь, что  $|B| < |A|$ . Тогда, поскольку  $f$  определено всюду на  $A$ ,  $\exists a_1, a_2 \in A$  такие, что  $f(a_1) = f(a_2) = b \in B$ . Снова получаем противоречие с тем, что  $f$  задает взаимно однозначное соответствие. Итак,  $|A| = |B|$ .

*Конечные и бесконечные множества.* Для дискретной математики  $\forall B \ (B \subset A \ \& \ |B| = |A|) \Rightarrow (B = A)$ .

Для конечного множества  $A$  используется обозначение  $|A| < \infty$ . Все остальные множества называются бесконечными. Взяв отрицание условия конечности для бесконечного множества, получаем:

$A \ni B \quad B \subset A \text{ \& } |B| = |A| \text{ \& } B \neq A$ , то есть бесконечное множество равномощно некоторому своему собственному подмножеству. Для бесконечного множества  $A$  используется обозначение  $|A| = \infty$ .

На компьютере все множества реальных объектов (множество адресуемых ячеек памяти, множество исполнимых программ, множество тактов работы процессора) конечны.

*Булеан.* Множество всех подмножеств множества  $M$  называется булеаном и обозначается  $2^M$ .  $2^M = \{A \mid A \subset M\}$ .

Для конечного множества  $M \mid 2^M| = 2^{|M|}$ .

*Пример 9.* Пусть  $A$  – множество всех натуральных четных чисел,  $B$  – множество всех натуральных чисел, представимых в виде суммы двух нечетных натуральных чисел. Докажем, что  $A = B$ .

$$A = \{2k \mid k \in \mathbb{N}\}, B = \{(2k-1) + (2m-1) \mid k, m \in \mathbb{N}\}.$$

Покажем, что для  $\forall x \in A \Rightarrow x \in B$  и для  $\forall y \in B \Rightarrow y \in A$ . Это будет означать, что  $A \subseteq B$  и  $B \subseteq A$ .

Пусть  $2k \in A$  для некоторого фиксированного  $k \in \mathbb{N}$ . Тогда  $2k = (2k-1) + 1$  – сумма двух нечетных натуральных чисел, поэтому  $2k \in B$ . Пусть  $(2k-1) + (2m-1) \in B$  для некоторых фиксированных  $k, m \in \mathbb{N}$ . Тогда  $(2k-1) + (2m-1) = 2k + 2m - 2 = 2(k+m-1)$  является четным натуральным числом, поэтому  $(2k-1) + (2m-1) \in A$ .

Итак,  $A = B$ .

*Пример 10.* Доказать, что множества  $X$  и  $Y$  равномощны, построив взаимно однозначное соответствие между ними.  $X = 3\mathbb{Z}$  (множество всех целых чисел, кратных 3),  $Y = \mathbb{N} \cup \{-1, -2\}$ .

*Доказательство.* Множество  $Y$  равномощно множеству натуральных чисел  $\mathbb{N}$ , т. к.  $Y$  есть сумма конечного множества и множества  $\mathbb{N}$ . Множество  $X$  равномощно множеству натуральных чисел  $\mathbb{N}$ , т. к. каждому натуральному числу  $i$  однозначно соответствует число  $3i$ . Тогда по теореме Кантора – Бернштейна [3] два множества, равномощные третьему, равномощны друг другу.

### 3.2. Операции над множествами. Алгебра множеств

*Объединением* множеств  $A$  и  $B$  (обозначение  $A \cup B$ ) называется множество, состоящее из всех тех и только тех элементов, которые принадлежат хотя бы одному из множеств  $A, B$ :  $A \cup B = \{x \mid (x \in A) \vee (x \in B)\}$ .

Аналогично определяется объединение произвольной (в том числе бесконечной) системы множеств. Если система содержит небольшое количество множеств, то их объединение описывается явно:  $A \cup B \cup C \cup D$  и т. д.

*Пример 11.* Пусть заданы множества  $A := \{a, b, d\}$  и  $B := \{b, d, e, h\}$ . Тогда  $A \cup B = \{a, b, d, e, h\}$ .

*Пример 12.* Пусть  $A := \{1, 2, 3\}$ ,  $B := \{3, 4, 5\}$ . Тогда  $A \cup B = \{1, 2, 3, 4, 5\}$ .

*Пример 13.* Обозначим  $N_k = \{n \in N \mid (k \text{ делит } n) \& (k \neq n)\}$ ,  $P$  – множество всех простых чисел (принято считать, что  $1 \notin P$ ). Тогда  $\bigcup_{i \in P} N_i$  – множество всех составных чисел.

Рассмотрим два круга, приведенных на рис. 1. Если  $A$  – множество точек левого круга,  $B$  – множество точек правого круга, то  $A \cup B$  представляет собой заштрихованную область, ограниченную обоими кругами.

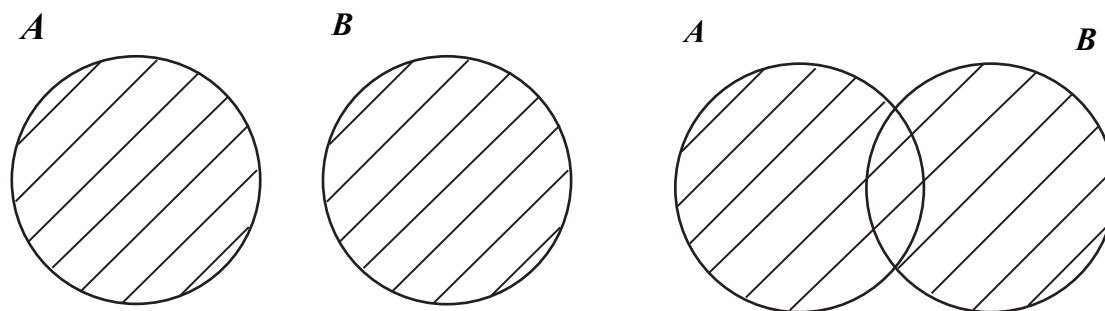


Рис. 1. Объединение множеств

Для объединения множеств существуют коммутативный и ассоциативный законы:  $A \cup B = B \cup A$ ,  $(A \cup B) \cup C = A \cup (B \cup C)$ , справедливость которых для любых множеств  $A$ ,  $B$  и  $C$  вытекает из того, что левая и правая части состоят из одних и тех же элементов. Рассмотрим  $A \cup \emptyset = A$ . Это соотношение верно для произвольного множества  $A$ , так как  $\emptyset$  не содержит элементов, а значит,  $A$  и  $A \cup \emptyset$  состоят из одних и тех же элементов. Очевидно, что  $A \cup A = A$  для любого множества  $A$ . Для любых множеств  $A$  и  $B$  верно  $A \subseteq (A \cup B)$ , но неверно  $A \in (A \cup B)$ .

*Пересечением* множеств  $A$  и  $B$  (обозначение  $A \cap B$ ) называется множество, состоящее из всех тех и только тех элементов, которые принадлежат и  $A$ , и  $B$ :  $A \cap B = \{x \mid (x \in A) \& (x \in B)\}$ .

Аналогично определяется пересечение (в том числе бесконечной) системы множеств. Обозначения для пересечения системы множеств аналогичны приведенным выше обозначениям для объединения.

*Пример 14.* Пусть  $A := \{a, b, d, c, e, f, k\}$ ,  $B := \{b, d, e, h\}$ . Тогда  $A \cap B = \{b, d, e\}$ .

*Пример 15.* Пусть  $A := \{1, 2, 3\}$ ,  $B := \{3, 4, 5\}$ . Тогда  $A \cap B = \{3\}$ .

Рассмотрим два круга, приведенных на рис. 2. Если  $A$  – множество точек левого круга,  $B$  – множество точек правого круга, то  $A \cap B$  представляет собой заштрихованную область, являющуюся общей частью этих кругов.

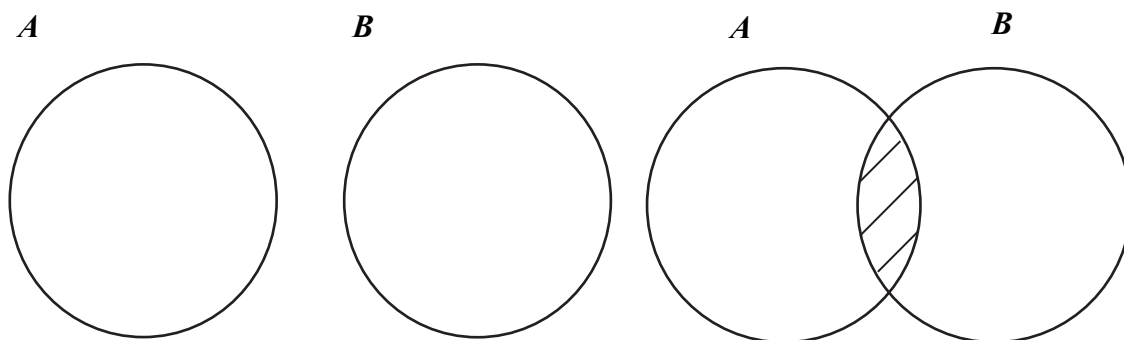


Рис. 2. Пересечение множеств

Нетрудно видеть, что пересечение заданных произвольных множеств обладает коммутативным ( $A \cap B = B \cap A$ ) и ассоциативным ( $(A \cap B) \cap C = A \cap (B \cap C)$ ) свойствами, соотношения верны для произвольных конечных множеств  $A$ ,  $B$  и  $C$ . Заметим также, что имеют место соотношения  $A \cap \emptyset = \emptyset$ ,  $A \cap A = A$ , верные для произвольного множества  $A$ .

*Разностью* множеств  $A$  и  $B$  (обозначение  $A \setminus B$ ) называется множество всех тех и только тех элементов  $A$ , которые не содержатся в  $B$ :  $A \setminus B = \{x \mid (x \in A) \& (x \notin B)\}$ . В отличие от операций объединения и пересечения, данная операция определяется только для двух множеств. Для произвольных множеств  $A$  и  $B$  верны соотношения:

$$A \setminus B = \emptyset \Leftrightarrow A \subseteq B, A \setminus \emptyset = A, A \setminus B = A \Leftrightarrow A \cap B = \emptyset.$$

*Пример 16.* Пусть заданы множества  $A := \{a, b, d\}$ ,  $B := \{b, d, e, h\}$ . Тогда  $A \setminus B = \{a\}$ ,  $B \setminus A = \{e, h\}$ . Таким образом, разность множеств не коммутативна.

*Пример 17.* Пусть  $A := \{1, 2, 3\}$ ,  $B := \{3, 4, 5\}$ . Тогда  $A \setminus B = \{1, 2\}$ .

*Пример 18.* Пусть  $A := \{a, b\}$ ,  $B := \{b, c\}$ ,  $C := \{a, c\}$ . Тогда  $A \setminus (B \setminus C) = A \setminus \{b\} = \{a\}$ .  $(A \setminus B) \setminus C = \{a\} \setminus C = \emptyset$ . Таким образом, разность множеств не ассоциативна в общем случае.

Рассмотрим два круга, приведенные на рис. 3. Если  $A$  – множество точек левого круга,  $B$  – множество точек правого круга, то  $A \setminus B$  представляет собой заштрихованную область.

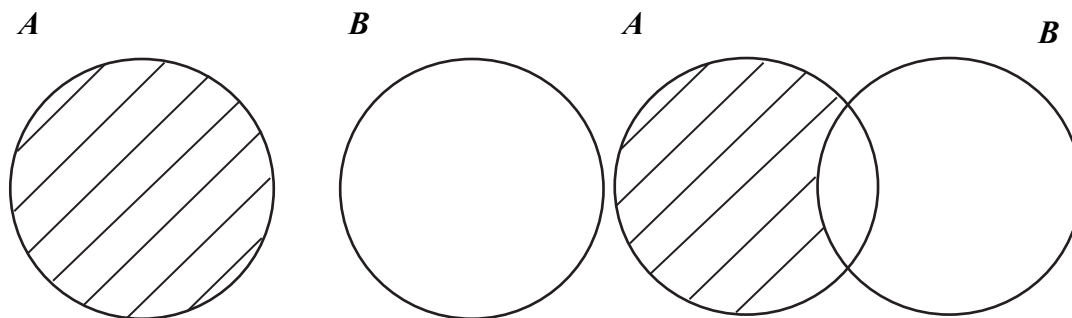


Рис. 3. Разность множеств

Если в некотором рассмотрении участвуют только подмножества фиксированного множества  $U$ , то это самое большое множество называют *полным* или *универсальным множеством*. Для любого множества  $A \subseteq U$ , очевидно, выполняются соотношения:  $A \cap U = A$ ,  $A \cup U = U$ .

Следует отметить, что в различных конкретных случаях роль универсального множества могут играть различные множества. Так, при рассмотрении множеств студентов в группе (отличники; студенты, получающие стипендию; студенты, проживающие в общежитии, и т. п.) роль универсального множества играет множество всех студентов данной группы.

Универсальное множество удобно изображать графически в виде множества точек прямоугольника. Отдельные области внутри этого прямоугольника будут обозначать различные подмножества универсального множества. Изображение множества в виде области в прямоугольнике, представляющем универсальное множество, называют диаграммой Эйлера – Венна.

*Дополнением* множества  $A$  до универсального множества  $U$  (обозначение  $\bar{A}$ ) называется множество всех элементов  $U$ , не принадлежащих  $A$ :  $\bar{A} = \{x \mid (x \in U) \ \& \ (x \notin A)\}$ ,  $\bar{A} = U \setminus A$ .

*Пример 19.* Если  $U := \{1, 2, 3, 4, 5, 6, 7\}$ ,  $A := \{3, 5, 7\}$ , то  $\bar{A} = \{1, 2, 4, 6\}$ .

На рис. 4 множество  $A$  показано заштрихованным кругом,  $\bar{A}$  представляет собой всю незаштрихованную область прямоугольника  $U$ .

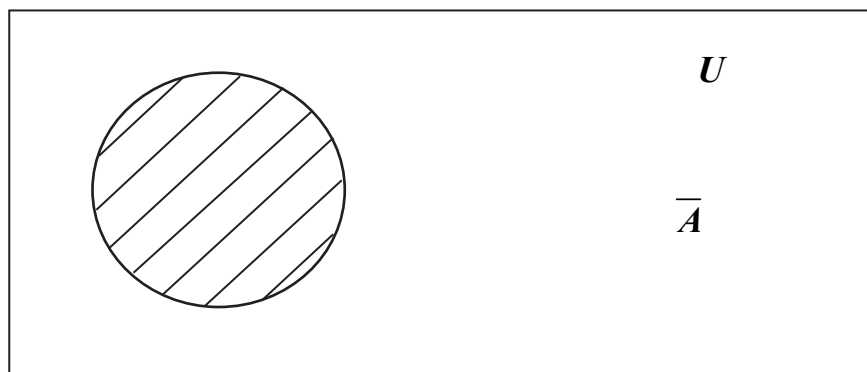


Рис. 4. Дополнение множества

Из определения дополнения множества следует, что  $A$  и  $\bar{A}$  не имеют общих элементов, так что  $A \cap \bar{A} = \emptyset$  для любого множества  $A$ . Кроме того, нет элементов  $U$ , которые не принадлежали бы ни  $A$ , ни  $\bar{A}$ , так как те элементы, которые не принадлежат  $A$ , принадлежат  $\bar{A}$  и наоборот. Следовательно,  $A \cup \bar{A} = U$  для любого множества  $A \subseteq U$ . Для любого множества  $A \subseteq U$  верно, что дополнением к  $\bar{A}$  является само множество  $A$ . Таким образом,  $\overline{\bar{A}} = A$ .

С помощью операции дополнения можно в удобном виде представить разность произвольных множеств  $A$  и  $B$ :

$$A \setminus B = \{x \mid (x \in A) \ \& \ (x \notin B)\} = \{x \mid (x \in A) \ \& \ (x \in \bar{B})\} = A \cap \bar{B}.$$

Наряду с понятием множества как совокупности элементов важным является понятие упорядоченного множества, или кортежа.

*Кортеж* – это последовательность элементов, т. е. совокупность элементов, в которой каждый элемент занимает определенное место. Часто кортеж называют вектором, а элементы, образующие кортеж, – его компонентами, или координатами. Компоненты нумеруются слева направо. Число компонент называется длиной, или размерностью, кортежа. Могут быть и бесконечные кортежи. В отличие от элементов множества, координаты кортежа могут совпадать. Кортеж будем заключать в круглые скобки. Например,  $a = (a_1, \dots, a_n)$  – кортеж длиной  $n$  с элементами  $a_1, \dots, a_n$ . Иногда скобки и даже запятые опускаются. Кортежи длиной 2 называют упорядоченными парами (или просто парами), кортежи длиной 3 –



тройками и т. д. В общем случае кортежи длиной  $n$  называют упорядоченными  $n$ -ками (или просто  $n$ -ками). Частным случаем является кортеж  $(a)$  длиной 1 и пустой кортеж длиной 0, обозначаемый  $()$ . Примеры кортежей: множество людей, стоящих в очереди; множество слов во фразе; числа, выражающие координаты долгот и широт точки на местности, и т. п.

Место каждого элемента в кортеже является вполне определенным и не может быть произвольно изменено. В технических задачах эта определенность часто является просто предметом договоренности. Два конечных кортежа равны, если они имеют одинаковую длину и соответствующие компоненты равны:

$$(a_1, \dots, a_m) = (b_1, \dots, b_n) \Leftrightarrow m = n \text{ и } a_1 = b_1, a_2 = b_2, \dots, a_m = b_m.$$

*Прямым произведением* множеств  $A$  и  $B$  (обозначение  $A \times B$ ) называется множество, состоящее из всех тех и только тех упорядоченных пар, первая компонента которых принадлежит множеству  $A$ , а вторая – множеству  $B$ .  $A \times B = \{(a, b) \mid a \in A, b \in B\}$ .

*Пример 20.* Пусть  $A := \{1, 2\}$ ,  $B := \{1, 3, 4\}$ .

Тогда:  $A \times B = \{(1, 1), (1, 3), (1, 4), (2, 1), (2, 3), (2, 4)\}$ ;

$B \times A = \{(1, 1), (1, 2), (3, 1), (3, 2), (4, 1), (4, 2)\}$ ;

$A \times B \neq B \times A$ .

Этот пример показывает, что прямое произведение множеств не коммутативно в общем случае.

Операция прямого произведения легко распространяется и на большее число множеств. Прямым произведением множеств  $A_1, A_2, \dots, A_r$ ,  $r \in N$ , называется множество, состоящее из всех тех и только тех кортежей длиной  $r$ , первая компонента которых принадлежит  $A_1$ , вторая  $A_2$ , ...,  $r$ -я –  $A_r$ .  $A_1 \times A_2 \times \dots \times A_r = \{(a_1, a_2, \dots, a_r) \mid a_i \in A_i, i = \overline{1, r}\}$ .

Частным случаем операции прямого произведения является понятие степени множества. Пусть  $A$  – произвольное множество. Назовем  $s$ -й степенью,  $s \in N$ , множества  $A$  (обозначение  $A^s$ ) прямое произведение  $s$  одинаковых множеств, равных  $A$ :  $A^s = \underbrace{A \times A \times \dots \times A}_{s \text{ раз}}$ .

Будем полагать, что  $A^1 = A$ ,  $A^0 = \{()\}$ .

Если  $R$  – вещественная прямая (множество вещественных чисел), то  $R^2 = R \times R$  представляет собой вещественную плоскость (множество

всех упорядоченных пар вида  $(a, b)$ , где  $a, b \in \mathbf{R}$ , которые являются координатами точек плоскости).  $\mathbf{R}^3 = \mathbf{R} \times \mathbf{R} \times \mathbf{R}$  – трехмерное вещественное пространство (множество всех упорядоченных троек вида  $(a, b, c)$ , где  $a, b, c \in \mathbf{R}$ , которые являются координатами точек пространства).

Координатное представление точек плоскости, предложенное французским математиком и философом Р. Декартом, – исторически первый пример прямого произведения. Поэтому иногда прямое произведение называют декартовым произведением.

Проекцией кортежа  $v = (v_1, \dots, v_n)$ ,  $n \in \mathbf{N}$ , на  $i$ -ю ось (обозначение  $\text{пр}_i v$ ) называется его  $i$ -я компонента. Проекцией кортежа  $v$  на оси с номерами  $i_1, \dots, i_k$ ,  $1 \leq i_1 < i_2 < \dots < i_k \leq n$ , называется кортеж  $(v_{i_1}, \dots, v_{i_k})$  длиной  $k$  (обозначение  $\text{пр}_{i_1, \dots, i_k} v$ ).

Операция проектирования множества тесно связана с операцией проектирования кортежа и может применяться лишь к таким множествам, элементами которых являются кортежи одинаковой длины. Пусть  $V$  – множество кортежей одинаковой длины. Тогда проекцией множества  $V$  на  $i$ -ю ось называется множество проекций всех векторов из  $V$  на  $i$ -ю ось:  $\text{пр}_i V = \{\text{пр}_i v \mid v \in V\}$ . Аналогично определяется проекция множества  $V$  на несколько осей:  $\text{пр}_{i_1, \dots, i_k} V = \{\text{пр}_{i_1, \dots, i_k} v \mid v \in V\}$ . В частности, если  $V = A_1 \times A_2 \times \dots \times A_n$ , то  $\text{пр}_i V = A_i$ ,  $i = \overline{1, n}$ ,  $\text{пр}_{i_1, \dots, i_k} V = A_{i_1} \times A_{i_2} \times \dots \times A_{i_k}$ ,  $1 \leq i_1 < i_2 < \dots < i_k \leq n$ . Если  $V \subseteq A_1 \times A_2 \times \dots \times A_n$ , то  $\text{пр}_i V \subseteq A_i$ ,  $i = \overline{1, n}$ ,  $\text{пр}_{i_1, \dots, i_k} V \subseteq A_{i_1} \times A_{i_2} \times \dots \times A_{i_k}$ .

Проекция точки плоскости на первую ось – это ее абсцисса (первая координата); проекция на вторую ось – ордината (вторая координата).

*Пример 21.* Для множества  $M := \{(x, y) \in \mathbf{R}^2 \mid (x - 2)^2 + y^2 = 1\}$  найти  $\text{пр}_1 M$  и  $\text{пр}_2 M$ .

*Решение.*  $M$  представляет собой окружность с центром в точке  $(2, 0)$ , радиусом 1 на плоскости (рис. 5). Из данного рисунка видно, что  $\text{пр}_1 M = \{x \in \mathbf{R} \mid x \in [1, 3]\}$  – проекция множества  $M$  на ось  $OX$ ,  $\text{пр}_2 M = \{y \in \mathbf{R} \mid y \in [-1, 1]\}$  – проекция множества  $M$  на ось  $OY$ .

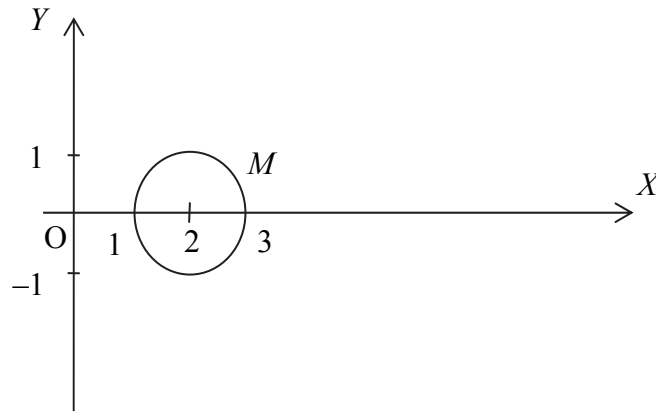


Рис. 5. Множество  $M$

Второй способ решения – аналитический.

$\text{Пр}_1 M = \{x \in \mathbf{R} \mid (x-2)^2 + y^2 = 1\} = \{x \in \mathbf{R} \mid (x-2)^2 = 1 - y^2\} = [y^2 \geq 0 \text{ для } \forall y \in \mathbf{R}] = \{x \in \mathbf{R} \mid (x-2)^2 \leq 1\} = \{x \in \mathbf{R} \mid -1 \leq x-2 \leq 1\} = \{x \in \mathbf{R} \mid 1 \leq x \leq 3\}$ .

$\text{Пр}_2 M = \{y \in \mathbf{R} \mid (x-2)^2 + y^2 = 1\} = \{y \in \mathbf{R} \mid y^2 = 1 - (x-2)^2\} = [(x-2)^2 \geq 0 \text{ для } \forall x \in \mathbf{R}] = \{y \in \mathbf{R} \mid y^2 \leq 1\} = \{y \in \mathbf{R} \mid -1 \leq y \leq 1\}$ .

И получаем тот же результат.

*Пример 22.* Пусть  $V := \{1, 2, 3, 4, 5\}, (2, 1, 3, 5, 5), (3, 3, 3, 3, 3), (3, 2, 3, 4, 3)\}$ . Тогда  $\text{пр}_2 V = \{2, 1, 3\}$ ,  $\text{пр}_{2,4} V = \{(2, 4), (1, 5), (3, 3)\}$ .

Пусть задано множество  $U$ ,  $P(U)$  – булеан множества  $U$ . Алгебра  $B = (P(U), \cup, \cap, -)$  называется булевой алгеброй множеств над  $U$ . Операции объединения, пересечения и дополнения часто называют булевыми операциями над множествами.

С помощью булевых операций из множеств можно составлять различные алгебраические выражения. Если оба алгебраических выражения представляют собой одно и то же множество, то их можно приравнять друг к другу, получив алгебраическое тождество. Часть основных тождеств булевой алгебры множеств была приведена при рассмотрении свойств булевых операций. Приведем доказательство остальных важных тождеств данной алгебры.

Пусть  $U$  – универсальное множество,  $A, B, C$  – произвольные подмножества  $U$ . На рис. 6 приведены диаграммы Эйлера – Венна для выражений  $(A \cup B) \cap C$  (пересечение заштрихованных множеств обозначено двойной штриховкой) и  $(A \cap C) \cup (B \cap C)$  (объединение заштрихованных множеств) соответственно. Их этих диаграмм видно, что оба выражения определяют одно и то же множество, так что в алгебре множеств имеет место тождество:  $(A \cup B) \cap C = (A \cap C) \cup (B \cap C)$ .

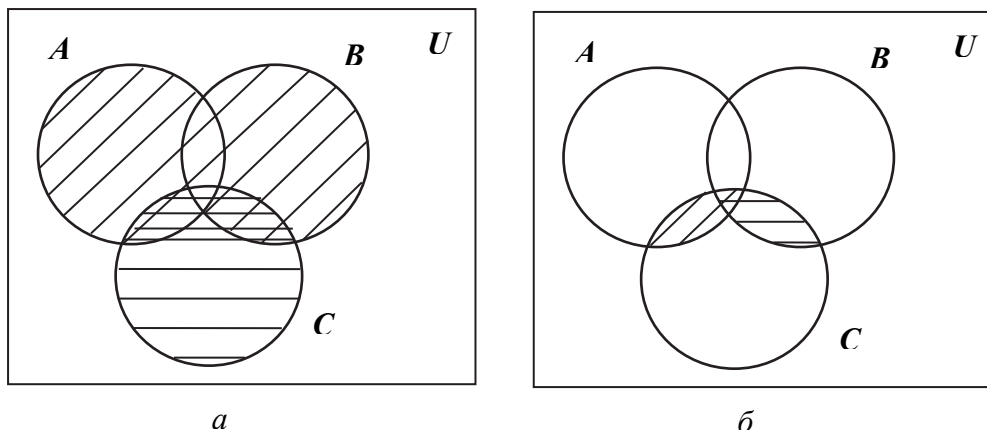


Рис. 6. Диаграммы Эйлера – Венна для  $(A \cup B) \cap C$  (а) и  $(A \cap C) \cup (B \cap C)$  (б)

Таким образом, пересечение дистрибутивно относительно объединения множеств.

На рис. 7 приведены диаграммы Эйлера – Венна для алгебраических выражений  $(A \cap B) \cup C$  (объединение заштрихованных множеств) и  $(A \cup C) \cap (B \cup C)$  (пересечение заштрихованных множеств обозначено двойной штриховкой) соответственно. Оба эти выражения дают одно и то же множество, так что имеет место тождество:

$$(A \cap B) \cup C = (A \cup C) \cap (B \cup C).$$

Таким образом, объединение дистрибутивно относительно пересечения множеств. Установление тождеств алгебры множеств с помощью диаграмм Эйлера – Венна в ряде случаев оказывается неудобным. Доказательство тождеств может производиться также методом двустороннего включения, чтобы показать равенство множеств в левой и правой частях тождества (об этом уже упоминалось выше), методом преобразования одной части к другой, методом преобразования обеих частей к одному и тому же выражению.

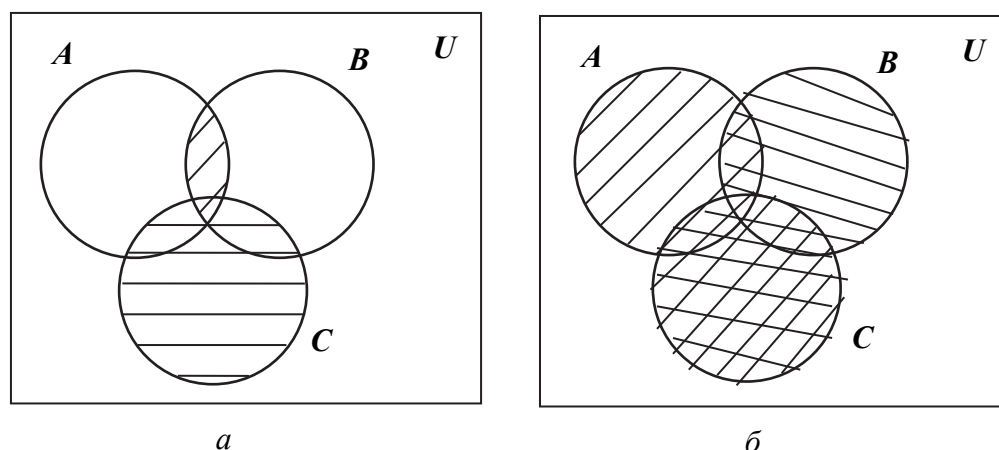


Рис. 7. Диаграммы Эйлера – Венна для  $(A \cap B) \cup C$  (а) и  $(A \cup C) \cap (B \cup C)$  (б)

Установление тождеств алгебры множеств с помощью диаграмм Эйлера – Венна в ряде случаев оказывается неудобным. Доказательство тождеств может производиться также методом двустороннего включения, чтобы показать равенство множеств в левой и правой частях тождества (об этом уже упоминалось выше), методом преобразования одной части к другой, методом преобразования обеих частей к одному и тому же выражению.

Пусть  $U$  – универсальное множество,  $A, B$  – его произвольные подмножества. Докажем тождество  $\overline{A \cup B} = \overline{A} \cap \overline{B}$ .

Будем использовать метод двустороннего включения. Предположим, что  $x \in \overline{A \cup B}$ , т. е. что  $x \notin A \cup B$ . Это значит, что  $x \notin A$  и  $x \notin B$ , т. е.  $x \in \overline{A}$  и  $x \in \overline{B}$ . Следовательно,  $x \in \overline{A} \cap \overline{B}$ . Итак,  $\overline{A \cup B} \subseteq \overline{A} \cap \overline{B}$ . Предположим теперь, что  $y \in \overline{A} \cap \overline{B}$ , т. е.  $y \in \overline{A}$  и  $y \in \overline{B}$ . Это значит, что  $y \notin A$  и  $y \notin B$ , т. е. что  $y \notin A \cup B$ . Следовательно,  $y \in \overline{A \cup B}$ . Итак,  $\overline{A} \cap \overline{B} \subseteq \overline{A \cup B}$ . Таким образом, тождество доказано.

Тождество  $\overline{A \cap B} = \overline{A \cup B}$  докажем, приведя его части к одинаковому виду. Выполним операцию дополнения над обеими частями. Слева получим  $\overline{\overline{A \cap B}} = A \cap B$ . Справа получим  $\overline{\overline{A \cup B}} = \overline{\overline{A} \cap \overline{B}} = A \cap B$ , согласно предыдущему тождеству. Итак,  $\overline{\overline{A \cap B}} = \overline{\overline{A \cup B}} = A \cap B$ . Справедливость тождества доказана.

В литературе два последних доказанных тождества обычно называются тождествами де Моргана.

*Симметрической разностью* множеств  $A$  и  $B$  (обозначение  $A \Delta B$ ) называется множество  $(A \setminus B) \cup (B \setminus A)$ . Очевидно, что  $B \Delta A = A \Delta B$  для любых множеств  $A$  и  $B$ , т. е. симметрическая разность коммутативна.

Докажем, что  $A \Delta B = (A \cup B) \setminus (A \cap B)$  для произвольных множеств  $A$  и  $B$ , т. е. докажем тождество  $(A \setminus B) \cup (B \setminus A) = (A \cup B) \setminus (A \cap B)$ .

Пусть  $A$  и  $B$  – произвольные подмножества некоторого универсального множества. На рис. 8 приведены диаграммы Эйлера – Венна для выражений  $(A \setminus B) \cup (B \setminus A)$  (объединение заштрихованных множеств  $A \setminus B$  и  $B \setminus A$ ) и  $(A \cup B) \setminus (A \cap B)$  (из объединения заштрихованных множеств  $A$  и  $B$  исключается множество с двойной штриховкой  $A \cap B$ ) соответственно. Из этих диаграмм видно, что оба выражения определяют одно и то же множество. Тем самым доказана справедливость тождества и получена равносильная формула для вычисления симметрической разности множеств.

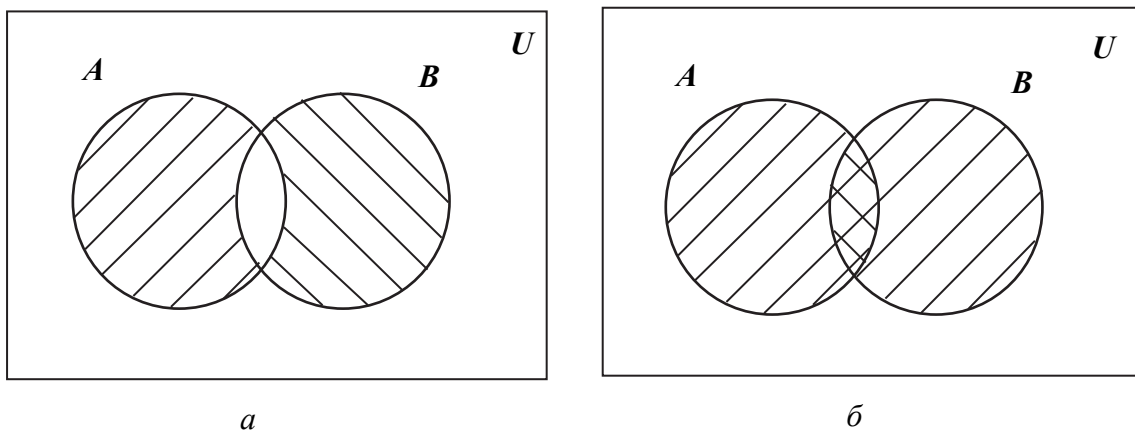


Рис. 8. Симметрическая разность

*Пример 23.* Доказать справедливость следующих тождеств:

- 1)  $A \setminus B = \overline{\overline{A} \cup B}$ ;
- 2)  $B = A \Delta (A \Delta B)$ ;
- 3)  $(A \cap B) \times C = (A \times C) \cap (B \times C)$ .

*Решение.*

1. Будем использовать метод преобразования правой части к левой. Преобразуем правую часть согласно одному из законов де Моргана:  $\overline{\overline{A} \cup B} = \overline{\overline{A}} \cap \overline{B} = A \cap \overline{B} = A \setminus B$ . Итак, тождество доказано.

2. Будем использовать метод диаграмм Эйлера – Венна для выражений в левой и правой частях. Пусть  $A$  и  $B$  – произвольные подмножества некоторого универсального множества  $U$ .

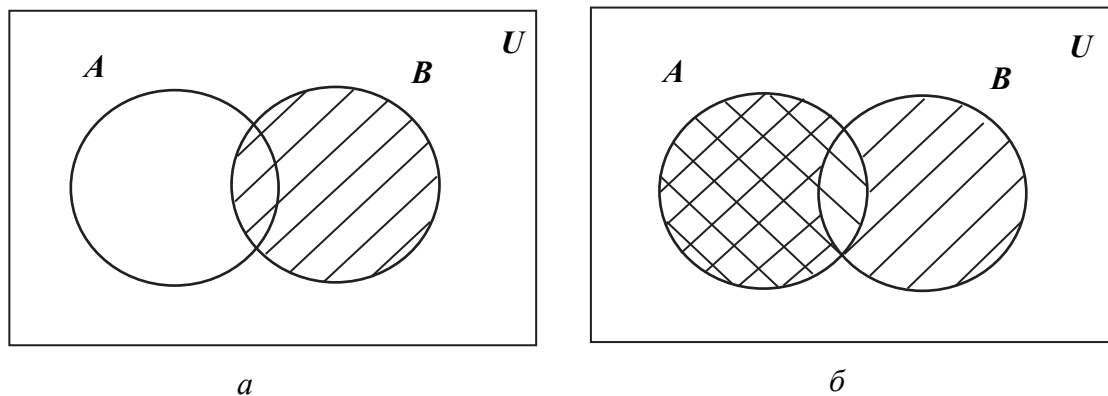


Рис. 9. Метод диаграмм Эйлера – Венна

На рис. 9 приведены диаграммы Эйлера – Венна для выражений  $B$  (заштрихованное множество) и  $A \Delta (A \Delta B)$  (отмеченное одинарной

штриховкой) следующим образом: сначала одной штриховкой было отмечено  $A \Delta B = (A \setminus B) \cup (B \setminus A)$ , затем другой штриховкой – множество  $A$ ; двойной штриховкой отмечено  $A \setminus B = A \cap (A \Delta B)$ . Из данных диаграмм видно, что оба выражения определяют одно и то же множество.

3. Докажем тождество методом двух взаимных включений.

$$(A \cap B) \times C = \{(x, y) \mid x \in A \cap B, y \in C\},$$

$$(A \times C) \cap (B \times C) = \{(x, y) \mid x \in A, y \in C\} \cap \{(x, y) \mid x \in B, y \in C\}.$$

Очевидно, что  $(A \cap B) \times C \subseteq (A \times C) \cap (B \times C)$  и  $(A \times C) \cap (B \times C) \subseteq (A \cap B) \times C$ . Итак, тождество доказано. Таким образом, прямое произведение дистрибутивно относительно пересечения множеств.

### 3.3. Соответствия. Отображения и функции. Отношения, их свойства и типы

Пусть  $X$  и  $Y$  – два непустых множества. Если определен способ сопоставления элементов  $Y$  элементам  $X$ , то говорят, что между множествами  $X$  и  $Y$  установлено соответствие. При этом совершенно не обязательно, чтобы в сопоставлении участвовали все элементы множеств  $X$  и  $Y$ . Для того чтобы задать соответствие между множествами  $X$  и  $Y$ , нужно задать множество  $Q \subseteq X \times Y$ , определяющее закон, по которому осуществляется соответствие, т. е. перечисляющий все пары  $(x, y)$ , участвующие в сопоставлении.

Таким образом, соответствие, обозначаемое  $q$ , представляет собой тройку множеств  $q = (X, Y, Q)$ , в которой  $Q \subseteq X \times Y$ . В этом выражении первую компоненту  $X$  называют областью отправления соответствия, вторую компоненту  $Y$  – областью прибытия соответствия, третью компоненту  $Q$  – графиком соответствия.

Кроме рассмотренных множеств  $X$ ,  $Y$  и  $Q$  с каждым соответствием неразрывно связаны еще два множества: множество  $\text{пр}_1 Q$ , называемое областью определения соответствия, которое состоит из всех элементов множества  $X$ , участвующих в сопоставлении, и множество  $\text{пр}_2 Q$ , называемое областью значений соответствия, которое состоит из всех элементов множества  $Y$ , участвующих в сопоставлении. Если  $(x, y) \in Q$ , то говорят, что элемент  $y$  соответствует элементу  $x$ . Геометрически это удобно изображать стрелкой, направленной от  $x$  к  $y$ . Если  $\text{пр}_1 Q = X$ , то соответствие называется всюду определенным, или отображением  $X$  в  $Y$  (в противном случае соответствие называется частичным). Если  $\text{пр}_2 Q = Y$ , то соответствие называется сюръективным (сюръекцией).

Множество всех  $y \in Y$ , соответствующих элементу  $x \in X$ , называется образом  $x$  в  $Y$  при соответствии  $q$ . Множество всех  $x \in X$ , которым соответствует элемент  $y \in Y$ , называется прообразом  $y$  в  $X$  при соответствии  $q$ . Если  $C \subseteq \text{пр}_1 Q$ , то образом множества  $C$  называется объединение образов всех элементов  $C$ . Аналогично определяется прообраз множества  $D$  для любого  $D \subseteq \text{пр}_2 Q$ . Соответствие  $q$  называется инъективным (инъекцией), если любые различные  $x_1$  и  $x_2$  из  $\text{пр}_1 Q$  имеют различные образы и любые различные  $y_1$  и  $y_2$  из  $\text{пр}_2 Q$  имеют различные прообразы при соответствии  $q$ .

Соответствие  $q$  называется функциональным (или однозначным), если образом любого элемента  $x \in \text{пр}_1 Q$  является единственный элемент  $y \in \text{пр}_2 Q$ . Соответствие  $q$  между множествами  $X$  и  $Y$  называется взаимно однозначным, или биективным, биекцией (иногда пишут «1-1-соответствие»), если оно всюду определено, сюръективно и инъективно. Однозначное отображение называется функцией. Функция является инъективной, если различным  $x_1$  и  $x_2$  из  $X$  соответствуют различные  $y_1$  и  $y_2$  из  $Y$ , и сюръективной, если она сюръективна как соответствие. Функция называется биективной, если она одновременно инъективна и сюръективна.

*Пример 24.* Пусть  $X = \{1, 2\}$ ,  $Y = \{3, 5\}$ , значит,  $X \times Y = \{(1, 3), (1, 5), (2, 3), (2, 5)\}$ . Это множество дает возможность получить 16 различных соответствий. Графики соответствий:

$$\begin{aligned}
 Q_0 &= \{(\ )\} = \emptyset; \\
 Q_1 &= \{(1, 3)\}; \\
 Q_2 &= \{(1, 5)\}; \\
 Q_3 &= \{(2, 3)\}; \\
 Q_4 &= \{(2, 5)\}; \\
 Q_5 &= \{(1, 3), (1, 5)\}; \\
 Q_6 &= \{(1, 3), (2, 3)\}; \\
 Q_7 &= \{(1, 3), (2, 5)\}; \\
 Q_8 &= \{(1, 5), (2, 3)\}; \\
 Q_9 &= \{(1, 5), (2, 5)\}; \\
 Q_{10} &= \{(2, 3), (2, 5)\}; \\
 Q_{11} &= \{(1, 3), (1, 5), (2, 3)\}; \\
 Q_{12} &= \{(1, 3), (1, 5), (2, 5)\}; \\
 Q_{13} &= \{(1, 3), (2, 3), (2, 5)\}; \\
 Q_{14} &= \{(1, 5), (2, 3), (2, 5)\}; \\
 Q_{15} &= \{(1, 3), (1, 5), (2, 3), (2, 5)\} = X \times Y.
 \end{aligned}$$



Обозначим  $q_i$  соответствие с графиком  $Q_i$ ,  $i = \overline{0, 15}$ . Рассмотрим соответствие  $q_9$ . Областью определения соответствия  $q_9$  является  $\text{пр}_1 Q_9 = \{1, 2\} = X$ , поэтому  $q_9$  – отображение. Областью значений  $q_9$  является  $\text{пр}_2 Q_9 = \{5\} \neq Y$ , поэтому  $q_9$  не сюръективно. Элемент 5 является образом 1 и 2 при соответствии  $q_9$ , поэтому  $q_9$  не инъективно. Образом множества  $X$  при соответствии  $q_9$  является множество  $\{5\}$ . Прообразом множества  $\{5\}$  при соответствии  $q_9$  является множество  $X$ . Соответствие  $q_9$  функционально, так как каждый из элементов 1 и 2 имеет единственный образ – элемент 5.

Отображениями являются соответствия  $q_6-q_9$ ,  $q_{11}-q_{15}$ . Сюръективными соответствиями являются  $q_5$ ,  $q_7$ ,  $q_8$ ,  $q_{10}-q_{15}$ . Функциональные соответствия:  $q_1-q_4$ ,  $q_6-q_9$ . Инъективные соответствия:  $q_1-q_4$ ,  $q_7$ ,  $q_8$ . Функциями являются  $q_6-q_9$ . Биактивные функции:  $q_7$ ,  $q_8$ .

Рассмотрим еще несколько примеров для приведенных выше определений.

Англо-русский словарь устанавливает соответствие между множествами английских и русских слов. Это соответствие не является функциональным, так как одному английскому слову, как правило, ставится в соответствие несколько русских слов. Кроме того, оно практически никогда не является полностью определенным: всегда можно найти английское слово, не содержащееся в данном словаре. Поскольку, как правило, всегда можно найти и русское слово, не содержащееся в данном словаре, соответствие не является сюръективным. При этом остается в стороне вопрос, является ли множество английских слов (также как и русских) точно заданным множеством.

Различные виды кодирования – кодирование букв азбукой Морзе, представления чисел в различных системах счисления, секретные шифры и т. д. – являются соответствиями между кодируемыми объектами и присваиваемыми им кодами. Эти соответствия, как правило, обладают всеми свойствами взаимно однозначного соответствия, кроме, быть может, одного – сюръективности. Инъективность следует из однозначности шифровки и дешифровки. Отсутствие сюръективности подразумевает, что не всякий код имеет смысл, т. е. соответствует какому-либо объекту.

Например, кодирование телефонов г. Минска семизначными номерами не сюръективно, так как некоторые семизначные номера не соответствуют никаким телефонам.

Множество всех векторов вида  $(n, 2^n)$ , где  $n \in \mathbb{N}$ , задает взаимно однозначное соответствие (биективную функцию) между множеством натуральных чисел и множеством всех натуральных степеней числа 2.

Композицией двух соответствий называется последовательное применение двух соответствий. Композиция двух соответствий есть операция с тремя множествами  $X, Y, Z$ , на которых определены два соответствия:

$$\left. \begin{aligned} q &= (X, Y, Q), \quad Q \subseteq X \times Y; \\ p &= (Y, Z, P), \quad P \subseteq Y \times Z. \end{aligned} \right\}$$

Причем область значений первого совпадает с областью определения второго соответствия:  $\text{пр}_2 Q = \text{пр}_1 P$ . Первое соответствие  $q$  определяет для любого  $x \in \text{пр}_1 Q$  некоторый, возможно и не один, элемент  $y \in \text{пр}_2 Q$ . Согласно определению операции композиции соответствий, теперь нужно для каждого такого  $y$  найти все соответствующие элементы  $z \in \text{пр}_2 P$ , воспользовавшись вторым соответствием  $p$ .

Композицию соответствий  $q$  и  $p$  будем обозначать  $p \circ q$  (знак  $\circ$  аналогичен умножению и часто опускается), а график композиции соответствий – через  $Q \circ P$ . При этом композиция соответствий  $p$  и  $q$  будет записана в виде:

$$p \circ q = (X, Z, Q \circ P), \quad Q \circ P \subseteq X \times Z.$$

Например, если  $q$  – соответствие, определяющее распределение шоферов по автомашинам,  $p$  – соответствие, определяющее распределение автомашин по маршрутам, то соответствие  $p \circ q$  есть соответствие, определяющее распределение шоферов по маршрутам.

Естественно, что операцию композиции можно распространить и на большее, чем два, число соответствий. Композиция  $n$  соответствий для  $n = 3, 4, 5, \dots$  определяется аналогично случаю  $n = 2$ .

Пусть  $X$  – множество людей. Для каждого человека  $x \in X$  обозначим через  $q(x)$  множество его детей. Тогда  $q^2(x)$  – множество внуков  $x$ ;  $q^3(x)$  – множество правнуков  $x$ ;  $q^{-1}(x)$  – множество родителей  $x$  и т. д. Изображая людей точками и рисуя стрелки, идущие из  $x$  в  $q(x)$ , получаем родословное, или генеалогическое, дерево (рис. 10).

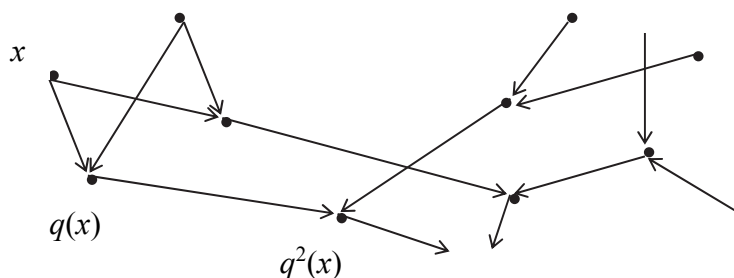


Рис. 10. Генеалогическое дерево

Пусть  $f: X \rightarrow Y$  – функция. Будем обозначать  $D(f)$  область определения функции и  $E(f)$  область значений функции  $f$ . Каждому элементу  $x \in X$   $f$  ставит в соответствие единственный элемент  $y \in Y$ . Это обозначается записью  $f(x) = y$  либо  $f: x \rightarrow y$ . Элемент  $x$  называется аргументом функции,  $y$  – значением функции на  $x$ . Если  $f(X)$  состоит из единственного элемента, то  $f$  называется функцией-константой. Тожественной функцией на множестве  $X$  называется функция  $e: X \rightarrow X$ , такая, что  $e(x) = x$  для любого  $x \in X$ . Если  $X, Y \subseteq \mathbf{R}$ , то функцию  $f$  называют вещественной.

Пусть даны функции  $f: X \rightarrow Y$  и  $g: Y \rightarrow Z$ . Функция  $h: X \rightarrow Z$  является композицией функций  $f$  и  $g$ ,  $h = g \circ f$ , если для любого  $x \in X$   $h(x) = g(f(x))$ . Часто говорят, что функция  $h$  получена подстановкой  $f$  в  $g$ .

Функция, полученная из  $f_1, \dots, f_n$  некоторой подстановкой их друг в друга и переименованием аргументов, называется суперпозицией  $f_1, \dots, f_n$ . Выражение, описывающее эту суперпозицию и содержащее функциональные знаки и символы аргументов, называется формулой.

Если  $f = (X, Y, Q_f)$ , то

$$Q_f = \{(x, y) \in X \times Y \mid y = f(x)\} = \{(x, f(x)) \in X \times Y\}.$$

Это соотношение позволяет установить способы задания функций.

1. Наиболее простой способ задания функций – это табличный. Таблицы при этом представляют собой конечные списки пар  $(x, f(x))$ . Однако таким способом могут быть заданы только функции, определенные на конечных множествах. Приведем примеры.

*Пример 25.* Из одного города в другой можно проехать по железной дороге, автобусом или катером. Стоимость билета будет соответственно 9000, 8000 и 10 000 руб. Стоимость билета можно представить как функцию от вида транспорта  $f: X \rightarrow Y$ , где  $X = \{\text{железная дорога, автобус, катер}\}$ ,  $Y = \{9000, 8000, 10\,000\}$ . Функция  $f$  может быть задана в виде таблицы.

$x$	Железная дорога	Автобус	Катер
$f(x)$	9 000	8 000	10 000

*Пример 26.* Рассмотрим множество  $X := \{1, 2, 3\}$  и два преобразования этого множества – функции  $\alpha$  и  $\beta$ ;  $\alpha: 1 \rightarrow 3, 2 \rightarrow 3, 3 \rightarrow 1$  и  $\beta: 1 \rightarrow 2, 2 \rightarrow 1, 3 \rightarrow 1$ ;  $\alpha$  и  $\beta$  могут быть заданы таблично:

$\alpha = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 3 & 1 \end{pmatrix}, \beta = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 1 \end{pmatrix}$ . Композиции преобразований также

можно задать таблично:  $\alpha\beta = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 3 & 3 \end{pmatrix}, \beta\alpha = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 1 & 2 \end{pmatrix}$ .

Отсюда видно, что  $\alpha\beta \neq \beta\alpha$ . Итак, данный пример показывает, что композиция функций, а в общем случае отображений и вообще соответствий, не коммутативна.

2. Другим не менее известным способом задания функций является аналитический, или формула, описывающая функцию с помощью суперпозиции других (исходных) функций. Если способ вычисления исходных функций известен, то формула задает процедуру вычисления данной функции как некоторую последовательность вычислений исходных функций.

*Пример 27.* Представить функцию в виде композиции исходных функций:

$$f(x) = \left( 1 + \left( \frac{x}{1-x} \right)^2 \right)^{\frac{1}{2}};$$

$$f_1(x) = \frac{x}{1-x}, D(f_1) = \mathbf{R} \setminus \{1\}, E(f_1) = \mathbf{R} \setminus \{-1\},$$

так как уравнение  $\frac{x}{1-x} = y$  разрешимо относительно  $x$  и  $x = \frac{y}{1+y}$  для любого  $y \in \mathbf{R} \setminus \{-1\}$ .

$$f_2(x) = x^2, D(f_2) = \mathbf{R}, E(f_2) = \mathbf{R}_{\geq 0}, E(f_2 f_1) = \mathbf{R}_{\geq 0};$$

$$f_3(x) = 1 + x, D(f_3) = \mathbf{R}, E(f_3) = \mathbf{R}, E(f_3 f_2 f_1) = \mathbf{R}_{\geq 1};$$

$$f_4(x) = x^{\frac{1}{2}}, D(f_4) = \mathbf{R}_{\geq 0}, E(f_4) = \mathbf{R}_{\geq 0}, E(f_4 f_3 f_2 f_1) = \mathbf{R}_{\geq 1}.$$

Видно, что  $E(f_i) \subset D(f_{i+1}), i = \overline{1, 3}$ .

Итак,  $f(x) = f_4(f_3(f_2(f_1(x)))) = f_4 f_3 f_2 f_1(x), D(f) = D(f_1) = \mathbf{R} \setminus \{1\}, E(f) = \mathbf{R}_{\geq 1}$ .

Пусть функция  $f = (\mathbf{R}, \mathbf{Q}_f)$ , где  $\mathbf{Q}_f = \{(x, x^2) \in \mathbf{R}^2\}$ . Тогда функцию  $f$  можно задать формулой  $f(x) = x^2, \forall x \in \mathbf{R}$ .

Иногда для разных подмножеств множества  $X$  при задании функции приходится пользоваться различными формулами. Пусть  $A_i \subset X, i = \overline{1, n}, A_i \cap A_j = \emptyset$  при  $i \neq j$ . Обозначим через  $f_i(x)$  формулу, определяющую  $y$  при  $x \in A_i, i = \overline{1, n}$ . Тогда функция  $f$ , определенная на всем множестве  $X$ , задается так:

$$f(x) = \begin{cases} f_1(x) & \text{при } x \in A_1; \\ f_2(x) & \text{при } x \in A_2; \\ \dots & \dots \\ f_n(x) & \text{при } x \in A_n. \end{cases}$$

*Пример 28.* Так, функцию  $f(x) = |x|$  можно задать в виде:

$$f(x) = \begin{cases} x & \text{при } x \geq 0; \\ -x & \text{при } x < 0. \end{cases}$$

3. Если  $f$  – вещественная функция, то элементы  $(x, f(x))$  можно изобразить в виде точек на плоскости  $\mathbf{R}^2$ . Полная совокупность таких точек будет представлять собой график функции  $f$ . Например, график функции  $f: [-1, 1] \rightarrow [0, 1]$ ,  $\mathcal{Q}_f = \{(x, y) \in \mathbf{R}^2 \mid x^2 + y^2 = 1\}$ ; график представлен на рис. 11.

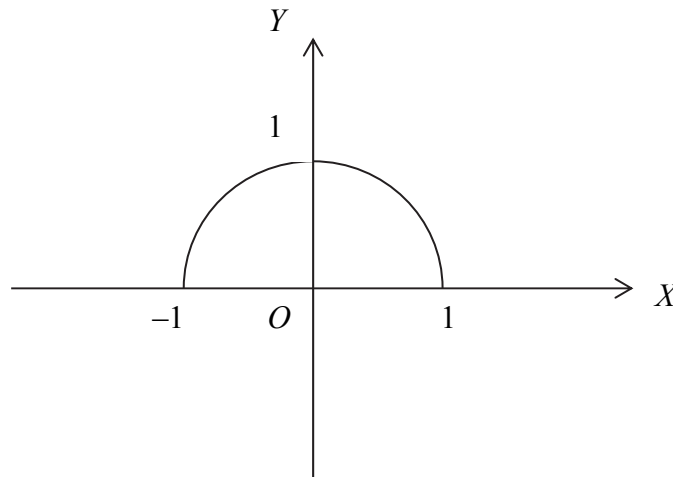


Рис. 11. График функции  $f$

4. Вычисления функций по таблицам, формулам, а также с помощью графиков являются частными видами вычислительных процедур. Существуют вычислительные процедуры, не относящиеся к указанным трем видам. Среди них особенно следует выделить рекурсивные процедуры. Рекурсивная процедура задает функцию  $f$ , определенную на множестве  $\mathbf{N} (\mathbf{N} \cup \{0\})$ , следующим образом:

- 1) задается значение  $f(1)$  ( $f(0)$ );
- 2) значение  $f(n+1)$  определяется через суперпозицию  $f(n)$  и других, считающихся известными, функций.

Простейшим примером рекурсивной процедуры является вычисление функции  $n!$ :

$$1) 0! = 1;$$

$$2) (n + 1)! = n!(n + 1).$$

Для вычисления  $(n + 1)!$  при  $n \in \mathbb{N}$  требуется  $(n - 1)$  умножений, т. е. число вычислительных шагов растет с ростом аргумента.

Приведем некоторые свойства функций и их композиций.

1. Композиция сюръективных функций сюръективна (следует из определений).

2. Композиция инъективных функций инъективна (следует из определений).

3. Композиция биективных функций биективна (следует из определений).

4. Композиция функций в общем случае не коммутативна.

*Пример 29.*  $f, g: \mathbb{R}_{>0} \rightarrow \mathbb{R}$ ,  $f(x) = \log_2 x$ ,  $g(x) = \log_3 x$ ;  $f$  и  $g$  – биекции. Покажем, что их композиция не коммутативна.

$$fg(2) = \log_2(\log_3 2) < 0, \text{ так как } \log_3 2 < 1;$$

$$gf(2) = \log_3(\log_2 2) = \log_3 1 = 0.$$

Поскольку  $fg(2) \neq gf(2)$ , то не может быть равенства  $fg = gf$  на всей области определения  $\mathbb{R}_{>0}$ .

5. Композиция функций ассоциативна.

Пусть  $f: X \rightarrow Y$ ,  $g: Y \rightarrow Z$ ,  $h: Z \rightarrow W$  – произвольные функции. Рассмотрим произвольный элемент  $x \in X$ .  $f(x) = y$ ,  $g(y) = z$ ,  $h(z) = w$ . Тогда  $hgf(x) = h(gf(x)) = h(g(y)) = h(z) = w$ ,  $(hg)f(x) = hg(f(x)) = hg(y) = h(g(y)) = h(z) = w$ ; т. е. для любого  $x \in X$   $h(gf)(x) = (hg)f(x)$ . Значит,  $h(gf) = (hg)f$ .

*Пример 30.* Даны три вещественные функции:

$$f(x) = 2x^9 - 7, g(x) = 5 \arctg(4x) + 2, h(x) = e^{5x} - 17.$$

Необходимо:

1) найти заданные композиции функций:  $fgh$ ,  $hfg$ ,  $ffg$ ;

2) узнать, являются ли заданные вещественные функции  $f, g, h$  инъекциями, сюръекциями, биекциями на  $\mathbb{R}$ ;

3) найти обратные функции к  $f, g, h$ . Если функции со своими областями определения обратных не имеют, то найти обратные функции к их сужениям.

1.  $D(f) = D(g) = D(h) = \mathbb{R}$ , поэтому все три указанные композиции функций определены на  $\mathbb{R}$ .

$$fgh(x) = f(gh(x)) = 2(gh(x))^9 - 7 = 2(5\operatorname{arctg}(4h(x)) + 2)^9 - 7 = \\ = 2(5\operatorname{arctg}(4(e^{5^x} - 17)) + 2)^9 - 7 = 2(5\operatorname{arctg}(4e^{5^x} - 68) + 2)^9 - 7.$$

$$hfg(x) = hf(g(x)) = hf(5\operatorname{arctg}(4x) + 2) = h(2(5\operatorname{arctg}(4x) + 2)^9 - 7) = \\ = e^{5^{2(5\operatorname{arctg}(4x) + 2)^9 - 7}} - 17.$$

$$ffg(x) = f(fg(x)) = f(2(5\operatorname{arctg}(4x) + 2)^9 - 7) = 2(2(5\operatorname{arctg}(4x) + 2)^9 - 7)^9 - 7.$$

2. Рассмотрим функцию  $f(x) = 2x^9 - 7$ . Производная функции  $f'(x) = 18x^8 > 0$  для всех  $x \in \mathbf{R} \setminus \{0\}$ , поэтому  $f$  является строго возрастающей функцией на  $(-\infty, 0) \cup (0, +\infty)$ ;  $f'(0) = 0$ ,  $f(0) = -7 \neq f(\xi)$  для  $\forall \xi \neq 0$ , поэтому  $f$  инъективна.

Функция  $f$  непрерывна на  $\mathbf{R}$ ;  $\lim_{x \rightarrow -\infty} f(x) = -\infty$  и  $\lim_{x \rightarrow +\infty} f(x) = +\infty$ , поэтому  $f$  является сюръекцией на  $\mathbf{R}$ . Итак,  $f: \mathbf{R} \rightarrow \mathbf{R}$  – биекция.

Рассмотрим функцию  $g(x) = 5\operatorname{arctg}(4x) + 2$ . Производная функции  $g$   $g'(x) = \frac{20}{1+16x^2} > 0$  для любого  $x \in \mathbf{R}$ . Следовательно, функция строго возрастает на всей области определения  $\mathbf{R}$ . Поэтому  $g$  – инъекция. Поскольку  $-\pi/2 < \operatorname{arctg}(4x) < \pi/2$  для всех  $x \in \mathbf{R}$ , то  $-5\pi/2 + 2 < g(x) < 5\pi/2 + 2$  для всех  $x \in \mathbf{R}$ . Значит,  $g$  не является сюръекцией на  $\mathbf{R}$ . Итак,  $g: \mathbf{R} \rightarrow \mathbf{R}$  не является биекцией.

Рассмотрим функцию  $h(x) = e^{5^x} - 17$ .  $h = h_3 h_2 h_1$ , где  $h_1(x) = 5^x$ ,  $h_2(x) = e^x$ ,  $h_3(x) = x - 17$ ,  $D(h_i) = \mathbf{R}$ ,  $i = 1, 2, 3$ ;  $h'_1(x) = \ln 5 \cdot 5^x > 0$  для  $\forall x \in \mathbf{R}$ , следовательно,  $h_1$  строго возрастает на  $\mathbf{R}$ , значит,  $h_1$  – инъекция.  $h'_2(x) = e^x > 0$  для  $\forall x \in \mathbf{R}$ , следовательно,  $h_2$  строго возрастает на  $\mathbf{R}$ , значит,  $h_2$  – инъекция. Пусть  $x_1 \neq x_2$ ; из равенства  $x_1 - 17 = x_2 - 17$  следует, что  $x_1 = x_2$ , поэтому  $h_3$  – инъекция. Функция  $h$  является инъективной функцией как композиция инъекций (по свойству 2 композиции функций).  $0 < 5^x < +\infty$  для всех  $x \in \mathbf{R}$ ,  $1 < e^{5^x} < +\infty$  для всех  $x \in \mathbf{R}$ ,  $-16 < e^{5^x} - 17 < +\infty$  для всех  $x \in \mathbf{R}$ . Значит,  $h$  не является сюръекцией на  $\mathbf{R}$ . Итак,  $h: \mathbf{R} \rightarrow \mathbf{R}$  не является биекцией.

3. Поскольку  $f: \mathbf{R} \rightarrow \mathbf{R}$  – биекция, то на  $\mathbf{R}$  существует обратная функция к  $f$  –  $f^{-1}: \mathbf{R} \rightarrow \mathbf{R}$ .

$$2x^9 - 7 = y;$$

$$x^9 = \frac{y+7}{2};$$

$$x = \sqrt[9]{\frac{y+7}{2}}.$$

$$\text{Итак, } f^{-1}(x) = \sqrt[9]{\frac{x+7}{2}}, x \in \mathbf{R}.$$

$E(g) = (-5\pi / 2 + 2, 5\pi / 2 + 2)$ . Поскольку  $g$  – инъекция на  $\mathbf{R}$ , то  $g: \mathbf{R} \rightarrow E(g)$  – биекция. Поэтому на  $E(g)$  существует обратная функция к  $g - g^{-1}: E(g) \rightarrow \mathbf{R}$ .

$$5\operatorname{arctg}(4x) + 2 = y;$$

$$5\operatorname{arctg}(4x) = y - 2;$$

$$\operatorname{arctg}(4x) = \frac{y-2}{5};$$

$$4x = \operatorname{tg}\left(\frac{y-2}{5}\right);$$

$$x = \frac{1}{4}\operatorname{tg}\left(\frac{y-2}{5}\right).$$

$$\text{Итак, } g^{-1}(x) = \frac{1}{4}\operatorname{tg}\left(\frac{x-2}{5}\right), x \in (-5\pi / 2 + 2, 5\pi / 2 + 2).$$

$E(h) = (-16; +\infty)$ . Поскольку  $h$  – инъекция на  $\mathbf{R}$ , то  $h: \mathbf{R} \rightarrow E(h)$  – биекция. Тогда для  $h$  существует обратная функция  $h^{-1}: E(h) \rightarrow \mathbf{R}$ .

$$e^{5x} - 17 = y;$$

$$e^{5x} = y + 17;$$

$$5x = \ln(y + 17);$$

$$x = \log_5(\ln(y + 17)).$$

$$\text{Итак, } h^{-1}(x) = \log_5(\ln(x + 17)), x \in (-16; +\infty).$$

Говоря об отображении  $f: X \rightarrow Y$  как о функции с вещественными значениями, на характер элементов множества  $X$  каких-либо особых ограничений не накладывали. Однако важным для практики является случай, когда множество  $X$  представляет собой множество функций, а множество  $Y$  – множество вещественных чисел. Этот случай приводит к понятию функционала, подробное рассмотрение которого удобно провести на примере.



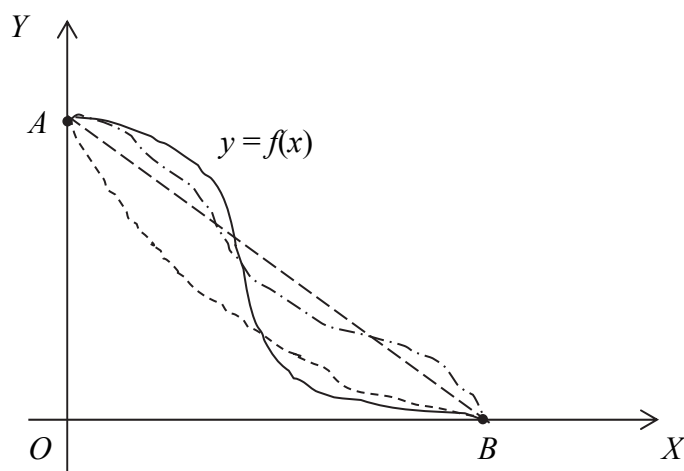


Рис. 12. График линий  $y = f(x)$

Представим себе некоторую линию  $y = f(x)$  в плоскости  $XOY$ , соединяющую фиксированные точки  $A$  и  $B$ , как показано на рис. 12, по которой скатывается свободно движущийся шарик. Обозначим через  $t$  время, которое шарик затратит на перемещение из точки  $A$  в точку  $B$ . Это время зависит от характера линии  $AB$ , т. е. от вида функции  $f(x)$ . Если обозначить через  $F(x)$  множество различных функций, изображающих линию  $AB$ , а через  $T$  – множество вещественных чисел  $t$ , определяющих время движения шарика, то зависимость времени движения от вида функции может быть записана как функциональное отображение:

$$j: F(x) \rightarrow T;$$

$$J = (F(x), T, J), \text{ где } J = \{(f(x), t) \mid f(x) \in F(x), t \in T\}.$$

В этом случае говорят, что вещественное число  $t \in T$  представляет собой значение функционала  $j$  от функции  $f(x) \in F(x)$ , записывают это в виде:

$$t = j[f(x)].$$

В задачах управления функционалы используются как критерии качества управления. Так, в рассмотренном примере время перемещения шарика из точки  $A$  в точку  $B$  можно трактовать как критерий «качества» выбранной функции  $f(x)$ . При этом говорят об оптимальном управлении как о таком, при котором соответствующий критерий качества обращается в минимум. С этой точки зрения определение «оптимального» вида функции  $f(x)$  сводится к выполнению условия  $\min_{f \in F} j[f(x)]$ , при кото-

ром время  $t$  будет минимальным. В математике подобная линия наискорейшего спуска получила название брахистохроны.

Оператором называется функциональное отображение  $l: X \rightarrow Y$ , в котором  $X$  и  $Y$  являются множествами функций одного аргумента  $t$ .

$$l = (X, Y, L),$$

где  $L = \{(x(t), y(t)) \in X \times Y\}$ ,  $L \subseteq X \times Y$ . В этом случае говорят, что оператор  $l$  преобразует функцию  $x(t)$  в функцию

$$y(t) = l[x(t)].$$

Примером оператора служит оператор дифференцирования  $p$ , ставящий в соответствие функции  $f(x)$  другую функцию  $f'(x) = \frac{df(x)}{dx}$ , что может быть записано в виде:

$$f'(x) = p[f(x)].$$

В задачах управления роль оператора часто выполняет сама управляемая система, преобразующая по некоторому закону  $l$  входной сигнал  $x(t)$  в выходной сигнал  $y(t)$ , как это показано на рис. 13.

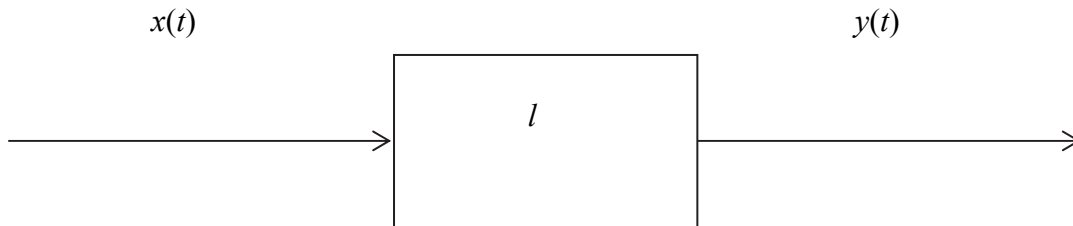


Рис. 13. Функциональное отображение

*Пример 31.* Рассмотрим предложенную фон Нейманом блок-схему ЭВМ, которая состоит из множества взаимосвязанных устройств  $M = \{a, b, c, d, e\}$ , где  $a$  – устройство ввода,  $b$  – арифметическое устройство (процессор),  $c$  – устройство управления,  $d$  – запоминающее устройство,  $e$  – устройство вывода.

Рассмотрим информационный обмен между устройствами  $m_i$  и  $m_j$ , которые находятся в отношении  $T(m_i, m_j)$ , если из устройства  $m_i$  поступает информация в устройство  $m_j$ . Опишем отношение  $T$  как множество упорядоченных пар:  $T \subset M^2$ .

$$T = \{(a, b), (a, c), (a, d), (b, c), (b, d), (b, e), (c, a), (c, b), (c, d), (c, e), (d, b), (d, c), (d, e), (e, c)\}.$$

Построим матрицу данного отношения:

$$C = \begin{matrix} & \begin{matrix} a & b & c & d & e \end{matrix} \\ \begin{matrix} a \\ b \\ c \\ d \\ e \end{matrix} & \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \end{matrix}.$$

*Бинарное отношение.* Бинарными отношениями пронизана вся математика и ее приложения: это отношение между двумя объектами.

Бинарное отношение можно определить как совокупность упорядоченных пар, указывающих объекты, находящиеся в данном отношении.

Например, если нас интересует отношение «уважать» между двумя – конкретными личностями  $a$  и  $b$ , то такое отношение может иметь различные формы. Так; если имеется отношение  $R = \{(a, b)\}$ , то это означает, что  $a$  уважает  $b$ . Отношение  $R_1 = \{(a, b), (b, a)\}$  означает, что  $a$  уважает  $b$  и  $b$  уважает  $a$ . Нетрудно интерпретировать также другие отношения «уважать» между интересующими нас лицами:  $R_2 = \{(a, b), (a, a)\}$ ,  $R_3 = \{(a, a), (b, b)\}$  и т. д.

В общем случае, если два элемента  $a, b$  находятся в данном отношении  $R$ , то этот факт записывают  $(a, b) \in R$  или  $aRb$ . Если эти элементы не находятся в отношении  $R$ , то это записывают так:  $(a, b) \notin R$ , или  $a\bar{R}b$ .

Некоторым из наиболее известных отношений присваивают специальные названия и обозначения. Примеры: эквивалентность ( $=$ ), отношение порядка ( $>$ ) или ( $>$ ), равенство ( $=$ ), параллельность ( $\parallel$ ), перпендикулярность ( $\perp$ ) и т. д.

Очевидно, что всякое бинарное отношение  $R$  можно рассматривать как подмножество прямого произведения некоторых множеств  $A$  и  $B$ :

$$R \subseteq A \times B.$$

Левой областью бинарного отношения называют множество всех первых компонент упорядоченных пар, составляющих данное отношение, то есть

$$R_- = \{a \mid (a, b) \in R\}.$$

Правой областью бинарного отношения  $R$  называют множество всех вторых компонент упорядоченных пар, составляющих данное отношение, то есть

$$R_+ = \{b \mid (a, b) \in R\}.$$

Например, пусть  $R = \{(1, 1), (1, 2), (1, 3), (3, 3)\}$ . Тогда  $R_- = \{1, 3\}$ ,  $R_+ = \{1, 2, 3\}$ .

Поле бинарного отношения  $R$  называют объединение его левой и правой областей:  $F(R) = R_- \cup R_+$ .

Бинарное отношение  $R^{-1}$  называют обратным к отношению  $R$ , если  $(a, b) \in R^{-1}$  тогда и только тогда, когда  $(b, a) \in R$ , то есть

$$R^{-1} = \{(a, b) \mid (b, a) \in R\}.$$

Например, если  $R = \{(1, 1), (1, 2), (1, 3), (3, 4)\}$ , то

$$R^{-1} = \{(1, 1), (2, 1), (3, 1), (4, 3)\}.$$

*Пересечением* бинарного отношения  $R$  по элементу  $a \in F(R)$  называют совокупность всех вторых (различных) компонентов упорядоченных пар, составляющих данное отношение, и таких, у которых первой компонентой есть элемент  $a$ . Обозначение:  $R_a$ .

Например, для предыдущего бинарного отношения  $R$  имеем:  $R_1 = \{1, 2, 3\}$ ,  $R_2 = \emptyset$ ,  $R_3 = \{4\}$ ,  $R_4 = \emptyset$ .

*Способы задания бинарных отношений.* Способы задания отношений зависят от свойств бинарного отношения. Различают следующие способы задания таких отношений.

1. Бинарное отношение  $R$  можно задать перечислением всех упорядоченных пар, находящихся в отношении  $R$ . Очевидно, что такой способ задания отношений приемлем для относительно небольшого числа упорядоченных пар. Например:  $R_1 = \{(1, 1), (2, 2), (3, 3), (4, 4)\}$ .

2. Если трудно перечислить все упорядоченные пары, составляющие отношение, то его можно задать формулой.

Например:  $S = \{(a, b) \mid (a - b) = 0 \bmod 3; a, b \in \{0..10\}\}$ .

3. Графическое задание бинарного отношения предполагает графическое представление элементов левой и правой областей отношения в виде точек в этих областях, соединенных дугами (направленными отрезками). Каждая дуга представляет некоторую упорядоченную пару, находящуюся в данном отношении. Дуга начинается в точке, соответствующей первой компоненте упорядоченной пары, и заканчивается в точке, соответствующей второй компоненте упорядоченной пары.

Пример отношения  $S = \{(a, b), (a, c), (b, c), (b, d)\}$  представлен на рис. 14.

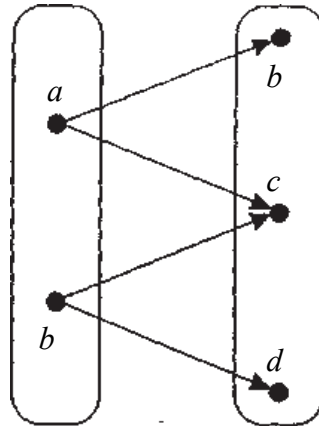


Рис. 14. Графическое представление бинарного отношения  $S$

4. Бинарное отношение можно задать в табличной форме. В этой таблице указываются все элементы поля отношения и соответствующие им пересечения данного отношения по выбранному элементу. Например:

$a$	$b$	$c$	$d$
$S_a = \{b, c\}$	$S_b = \{c, d\}$	$S_c = \emptyset$	$S_d = \emptyset$

5. Бинарное отношение можно задать матрицей  $\|a_{ij}\|$ , в которой строки и столбцы соответствуют полю отношения. В этой матрице  $i$ -я строка соотносится с некоторым элементом левой области отношения, а  $j$ -й столбец — с некоторым элементом правой области отношения. Тогда  $a_{ij} = 1$ , если соответствующие элементы находятся в данном отношении, и  $a_{ij} = 0$  в противном случае.

Например, отношение  $S$  из предыдущего пункта может быть представлено следующей таблицей.

	$a$	$b$	$c$	$d$
$a$	0	1	1	0
$b$	0	0	1	1
$c$	0	0	0	0
$d$	0	0	0	0

*Операции над бинарными отношениями.* Так как всякое бинарное отношение — это множество упорядоченных пар, то над бинарными отношениями можно выполнять все теоретико-множественные операции: объединение, пересечение, разность, дополнение.

Если  $R$  – бинарное отношение, то в качестве универсального множества в этом случае рассматривают множество  $U = F(R) \times F(R)$ , где  $F(R)$  – поле отношения  $R$ . Если совместно рассматривается несколько бинарных отношений, то в качестве универсального множества рассматривают множество  $U = A \times A$ , где  $A$  есть объединение полей каждого из рассматриваемых отношений.

Например, пусть рассматриваются отношения  $R = \{(1, 1), (1, 2), (1, 3), (3, 3)\}$  и  $S = \{(1, 1), (2, 2), (3, 3)\}$ . В этом случае универсальное множество имеет вид:  $U = \{(1, 1), (1, 2), (1, 3), (2, 1), (2, 2), (2, 3), (3, 1), (3, 2), (3, 3)\}$ .

Тогда результаты некоторых теоретико-множественных операций будут следующими:

$$\bar{R} = \{(2, 1), (2, 2), (2, 3), (3, 1), (3, 2)\};$$

$$\bar{S} = \{(1, 2), (1, 3), (2, 1), (2, 3), (3, 1), (3, 2)\}.$$

$$R \setminus S = \{(1, 2), (1, 3)\};$$

$$R \cap S = \{(1, 1), (3, 3)\}.$$

Кроме обычных теоретико-множественных операций, над бинарными отношениями определяют специальную операцию.

*Композицией* бинарных отношений  $R$  и  $S$  называют бинарное отношение  $T$ , состоящее из всех упорядоченных пар  $(a, b)$ , для каждой из которых существует элемент  $\tilde{c} \in \hat{E}_+ \cap S$  такой, что  $(a, c) \in R$ ,  $(c, b) \in S$  (то есть  $aRc$ ,  $cSb$ ). Операцию композиции записывают так:  $T = R \circ S$ .

Например, пусть

$$R = \{(1, 1), (1, 2), (2, 3), (3, 3)\}, S = \{(2, 4), (2, 5), (3, 2), (5, 5)\}.$$

$$\text{Тогда } R \circ S = \{(1, 4), (1, 5), (2, 2), (3, 2)\}, S \circ R = \{(3, 3)\}.$$

*Свойства бинарных отношений.* Бинарное отношение  $R$  называют рефлексивным, если для любого элемента  $a \in F(R)$  имеет место  $aRa$ .

Примерами рефлексивных отношений могут служить отношение подобия ( $\sim$ ), отношение параллельности ( $\parallel$ ), диагональное отношение на множестве  $A = \{a, b, c\}$ :

$$\Delta_A = \{(a, a), (b, b), (c, c)\}.$$

Бинарное отношение  $R$  называют антирефлексивным, если для любого элемента поля  $a \in F(R)$  имеет место  $a\bar{R}a$ .

Примерами антирефлексивных отношений являются отношения порядка ( $<$ ), ( $>$ ), отношение перпендикулярности.

Если задано бинарное отношение  $R_1 = \{(a, a), (a, b), (b, b), (a, c), (c, c)\}$ , то это отношение рефлексивно, а бинарное отношение  $R = \{(a, b), (b, c), (b, b), (a, c)\}$  – нет.

Бинарное отношение  $R_3 = \{(a, b), (b, c), (a, c)\}$  антирефлексивно.

Бинарное отношение  $R$  симметричное, если из  $aRb$  следует  $bRa$ .

Примерами таких отношений являются отношение равенства ( $=$ ), подобия ( $\sim$ ), диагональное отношение  $\Delta_A$ , отношение перпендикулярности, отношение параллельности ( $\parallel$ ). Бинарное отношение  $R$  асимметрично, если из  $aRb$  следует  $b\bar{R}a$ .

Асимметричными являются отношения порядка ( $<$ ), ( $>$ ).

Бинарное отношение  $R$  называют антисимметричным, если из  $aRb$  и  $bRa$  следует, что  $a = b$ . Заметим, что антисимметричное отношение отличается от асимметричного лишь тем, что в антисимметричном отношении допускается существование упорядоченной пары с одинаковыми компонентами.

Так, заданные бинарные отношения  $S_1 = \{(a, a), (b, b), (c, c)\}$  и  $S_2 = \{(a, a), (a, b), (a, c), (b, a), (c, a)\}$  симметричны.

С другой стороны, бинарные отношения  $S_1 = \{(a, a), (b, b), (c, c)\}$ ,  $S_3 = \{(a, b), (a, c), (a, a), (b, c)\}$  антисимметричны.

Бинарное отношение  $R$  называют транзитивным, если из  $aRb$  и  $bRc$  следует  $aRc$ .

Примерами транзитивных отношений являются отношение равенства ( $=$ ), отношение подобия ( $\sim$ ), диагональное отношение  $\Delta_A$ , отношения порядка, отношение параллельности ( $\parallel$ ). Примерами транзитивных отношений также могут служить отношения  $S_1$  и  $S_3$ .

В противном случае отношение  $R$  называют нетранзитивным.

В зависимости от свойств, которыми обладают бинарные отношения, выделяют и исследуют различные типы отношений. Наиболее известные из них – отношения эквивалентности и порядка.

Бинарное отношение называют отношением эквивалентности, если оно рефлексивно, симметрично и транзитивно.

Примеры отношения эквивалентности: отношение равенства ( $=$ ), отношение параллельности ( $\parallel$ ) и тому подобное.

*Классом эквивалентности  $R_a$*  называют множество всех вторых компонентов упорядоченных пар отношения эквивалентности  $R$ , у которых первой компонентой является элемент  $a$ :

$$R_a = \{b \mid (a, b) \in R\}.$$

Иначе говоря, классом эквивалентности называют пересечение отношения эквивалентности по элементу поля этого отношения.

Так, например, в отношении равенства класс эквивалентности любого элемента  $a$  состоит из единственного элемента  $a$ , класс эквивалентности отношения параллельности состоит из всех параллельных прямых.

Бинарное отношение  $S_1$  является отношением эквивалентности. В нем каждый класс эквивалентности состоит также из одного элемента:  $S_1(a) = \{a\}$ ,  $S_1(b) = \{b\}$  и  $S_1(c) = \{c\}$ .

Пусть имеется бинарное отношение

$$S = \{(a, a), (a, b), (b, a), (b, b), (c, c), (d, d), (d, e), (e, d), (e, e)\}.$$

Нетрудно видеть, что данное отношение рефлексивно, симметрично и транзитивно. Следовательно, отношение  $S$  есть отношение эквивалентности.

Имеем классы эквивалентности:

$$S_a = \{a, b\}, S_b = \{a, b\}, S_c = \{c\}, S_d = \{d, e\}, S_e = \{d, e\}.$$

Другими словами, имеется три различных класса эквивалентностей, порожденных данным отношением. С «точки зрения» свойств этого отношения, элементы  $a, b$  являются неразличимыми, как и элементы  $d, e$ .

*Диаграмма Хассе.* Для графического представления упорядоченного множества  $R$  используют диаграмму Хассе. Эта диаграмма строится следующим образом.

Каждому элементу поля  $F(R)$  ставится в соответствие точка (кружок) на плоскости, причем, если  $aRb$ , точку, соответствующую элементу  $a$ , располагают ниже точки, соответствующей элементу  $b$ . Точки, принадлежащие полю отношения порядка, то есть  $a, b \in F(R)$ , соединяют линией (ребром), если  $aRb$  и не существует элемента  $c \in F(R)$  такого, что  $aRc$  и  $cRb$ .

*Пример 32.* Пусть имеется отношение порядка:

$$R = \{(1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (1, 8), (2, 5), (2, 7), (2, 8), (3, 5), (3, 6), (3, 8), (4, 6), (4, 7), (4, 8), (5, 8), (6, 8), (7, 8)\}.$$

Диаграмма Хассе данного отношения представлена на рис. 15.

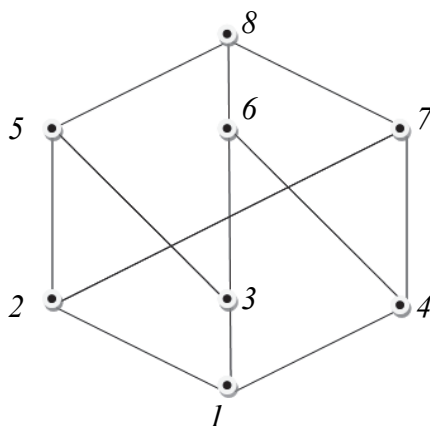


Рис. 15. Диаграмма Хассе



### 3.4. Алгоритмы упорядочивания множества

Пусть имеется неупорядоченное множество  $A = \{a_1, a_2, \dots, a_n\}$ , элементы которого записаны в памяти компьютера как элементы массива  $A = (a[1], a[2], \dots, a[n])$ , где  $a[i]$  есть содержимое ячейки с номером  $i$  ( $i = 1, 2, \dots, n$ ). Первоначально полагают, что  $a_i = a[i]$ .

В некоторых приложениях требуется упорядочить запись элементов в ячейках массива так, чтобы выполнялось соотношение  $a[i] < a[j]$ , если  $i < j$ , где  $i, j$  – номера ячеек массива. В общем случае здесь символ  $<$  определяет отношение порядка, т. е. необязательно отношение сравнения чисел. Это могут быть слова в некотором алфавите или иное. Однако для упрощения изложения будем предполагать, что элементами массива являются целые числа. Процедуру упорядочивания элементов массива называют сортировкой.

Строго говоря, рассматривая элементы множества  $A$  как массив, тем самым устанавливают порядок рассмотрения элементов этого множества в зависимости от номера ячейки, где они расположены. Тогда сущность сортировки состоит в перезаписи элементов массива в соответствии с заданным отношением порядка  $<$ .

Существует ряд известных методов упорядочения, каждый из которых предназначен для решения данной задачи. Разнообразие методов упорядочивания определяется потребностью применения наиболее эффективного метода для конкретного случая. Часто эффективность метода определяется временем, необходимым для сортировки массива. Время сортировки напрямую зависит от числа операций (сравнений, перезаписи элементов массива), которые необходимо выполнить согласно принятому методу.

Отсортированный массив удобен для дальнейшего использования, так как позволяет эффективно выполнять некоторые другие функции, например, вставку нового элемента в существующий отсортированный массив.

*Пузырьковая сортировка.* Сущность пузырьковой сортировки состоит в следующем. Пусть необходимо исходный массив чисел  $a[1], a[2], \dots, a[n]$  расположить в неубывающем порядке, т. е. так, чтобы выполнялось условие  $a[i] < a[j]$ , если  $i < j$ .

Будем рассматривать элементы массива как вектор-столбец, в первой строке которого расположен элемент  $a[1]$ , а в последней строке – элемент  $a[n]$  исходного массива.

Метод состоит в том, чтобы на каждой итерации метода наиболее «легкий» (с меньшим значением) элемент оказался «выше» всех более

«тяжелых» элементов массива. Из такого представления и проистекает название метода.

Каждая  $k$ -я итерация начинается с рассмотрения последнего,  $n$ -го элемента массива. В общем случае при рассмотрении  $i$ -го элемента массива происходит его сравнение с  $i - 1$ -м элементом. Если  $a[i - 1] > a[i]$ , то элементы массива меняются местами. В противном случае расположение элементов не меняется. Далее происходит переход к рассмотрению  $i - 1$ -го элемента массива. Итерация заканчивается при сравнении  $a[k + 1]$ -го элемента с элементом  $a[k]$ . В результате выполнения каждой итерации один, самый «легкий», элемент оказывается «наверху» рассматриваемой части вектора-столбца.

Сортировка заканчивается, когда в процессе ее выполнения не будет ни одной замены местами элементов массива.

*Пример 33.* Пузырьковым методом отсортировать массив целых чисел  $A = \{8, 4, 6, 3, 7, 2, 1, 5\}$ .

Изменения исходного массива в процессе сортировки представлены в виде последовательности вектор-столбцов, каждый из которых содержит результат соответствующей итерации (первый вектор-столбец представляет исходный массив).

$$\begin{pmatrix} 8 \\ 4 \\ 6 \\ 3 \\ 7 \\ 2 \\ 1 \\ 5 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 \\ 8 \\ 4 \\ 6 \\ 3 \\ 7 \\ 2 \\ 5 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 \\ 2 \\ 8 \\ 4 \\ 6 \\ 3 \\ 7 \\ 5 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 \\ 2 \\ 3 \\ 8 \\ 4 \\ 6 \\ 7 \\ 5 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 8 \\ 6 \\ 7 \\ 5 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 8 \\ 6 \\ 7 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 8 \\ 7 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{pmatrix}$$

Определим число операций, которые необходимо выполнить (в «худшем» случае) для получения отсортированного массива.

На первой итерации требуется, очевидно, выполнить  $n - 1$  сравнений и, возможно, столько же перемен местами сравниваемых элементов. На второй итерации уже потребуется выполнить  $n - 2$  сравнений и столько же перемен местами элементов. Легко заметить, что на  $k$ -й итерации потребуется выполнить  $n - k$  сравнений и столько же перемен местами элементов. Значит, всего нужно выполнить операций:

$$\sum_{k=1}^{n-1} 2(n - k) = n^2 - n.$$

*Сортировка выбором.* Идея метода сортировки с выбором в некоторой степени напоминает сортировку пузырьковым методом. Вначале неотсортированным полагают весь исходный массив. В процессе сортировки находится наименьший элемент, содержащийся в рассматриваемой (неотсортированной) части массива, который помещается взамен его первого элемента. Затем процедура повторяется – рассматривается неотсортированная часть массива, которая уменьшена по сравнению с предыдущей итерацией на один элемент.

Таким образом, на  $k$ -й итерации, как и в пузырьковой сортировке, рассматривают массив  $a[k], \dots, a[n]$  и, найдя в нем наименьший элемент, меняют его местами с элементом  $a[k]$ . Следовательно, сортировка заканчивается после выполнения 1-й итерации.

*Пример 34.* Сортировкой выбором упорядочить массив  $A = \{8, 4, 6, 3, 7, 2, 1, 5\}$ .

Изменения исходного массива в процессе сортировки представлены в виде последовательности массивов, каждый из которых есть результат соответствующей итерации:

(8, 4, 6, 3, 7, 2, 1, 5)	исходный массив;
(1, 4, 6, 3, 7, 2, 8, 5)	1-я итерация;
(1, 2, 6, 3, 7, 4, 8, 5)	2-я итерация;
(1, 2, 3, 6, 7, 4, 8, 5)	3-я итерация;
(1, 2, 3, 4, 7, 6, 8, 5)	4-я итерация;
(1, 2, 3, 4, 5, 6, 8, 7)	5-я итерация;
(1, 2, 3, 4, 5, 6, 7, 8)	6-я итерация.

Определим число операций, необходимых для сортировки массива. На первой итерации требуется, очевидно, выполнить  $n - 1$  сравнений и две операции перемен двух элементов местами. На второй итерации уже требуется выполнить  $n - 2$  сравнений и две операции перемещений элементов. Легко заметить, что на  $k$ -й итерации необходимо выполнить  $n - k$  сравнений и снова две операции перемен элементов местами. Значит, всего выполняется операций:

$$\sum_{k=1}^{n-1} (n - k + 2) = \frac{n^2 + 3n}{2} - 2.$$

*Сортировка вставками.* Сортировка вставками в некотором смысле является комбинацией двух предыдущих методов. Как при сортировке выбором, в процессе работы в начальной части массива формируется уже упорядоченная последовательность элементов и, как и в пузырьковой сортировке, поиск места вставки нового элемента в

отсортированную часть массива происходит последовательным сравнением этого элемента с элементами отсортированной части массива. Таким образом, на первом шаге метода полагаем, что первый элемент массива уже отсортирован.

В общем случае на  $k$ -й итерации метода имеем отсортированную часть массива  $a[1], \dots, a[k]$  и неотсортированную часть массива  $a[k+1], \dots, a[n]$ . К первой части присоединяется новый элемент  $a[k+1]$  и происходит поиск места его расположения сравнением с элементом  $a[k]$  (как при пузырьковой сортировке).

Если  $a[k] > a[k+1]$ , то эти элементы меняются местами и т. д. Итерация заканчивается, когда в результате сравнения не происходит перемены мест этих элементов.

Очевидно, что необходимо выполнить  $n - 1$  итераций.

*Пример 35.* Сортировкой вставками упорядочить массив

$$A = \{8, 4, 6, 3, 7, 2, 1, 5\}.$$

Как и ранее, изменения исходного массива в процессе сортировки представлены в виде последовательности массивов, каждый из которых есть результат соответствующей итерации:

(8, 4, 6, 3, 7, 2, 1, 5)	исходный массив;
(4, 8, 6, 3, 7, 2, 1, 5)	1-я итерация;
(4, 6, 8, 3, 7, 2, 1, 5)	2-я итерация;
(3, 4, 6, 8, 7, 2, 1, 5)	3-я итерация;
(3, 4, 6, 7, 8, 2, 1, 5)	4-я итерация;
(2, 3, 4, 6, 7, 8, 1, 5)	5-я итерация;
(1, 2, 3, 4, 6, 7, 8, 5)	6-я итерация;
(1, 2, 3, 4, 5, 6, 7, 8)	7-я итерация.

Определим число операций, которые необходимо выполнить (в «худшем» случае) для получения отсортированного массива.

На первой итерации требуется, очевидно, выполнить одно сравнение и, возможно, одну операцию перемен местами сравниваемых элементов. На второй итерации уже потребуется выполнить два сравнения и столько же изменений местами элементов.

Легко заметить, что на  $k$ -й итерации ( $k = 1, \dots, n - 1$ ) потребуется выполнить  $k$  сравнений и столько же перемен местами элементов. Значит, всего нужно выполнить операций:

$$\sum_{k=1}^{n-1} (2k) = 2 \frac{n(n-1)}{2} = n(n-1).$$

*Метод Шелла.* Метод Шелла представляет собой усовершенствование метода вставок за счет возможности обмена местами элементов, которые в исходном массиве находятся на значительном расстоянии друг от друга. Сущность метода состоит в том, что на каждой итерации элементы разделены на группы, в каждой из которых они упорядочиваются методом вставок.

Первоначально имеется большое число групп, соседние элементы в которых находятся в исходном массиве на большом расстоянии друг от друга. В полученных группах элементы упорядочиваются методом вставок. С каждой итерацией число групп уменьшается, объем их растет, но при этом уменьшается расстояние между соседними элементами группы, которое они имеют в текущем массиве. На последней итерации текущий массив образует одну группу, элементы которой также упорядочиваются методом вставок.

Заметим, что вышеупомянутые группы не требуют дополнительной памяти, формирование элементов в группы определяется лишь порядком сравнения элементов в текущей итерации.

На каждой итерации метода Шелла выбирается расстояние (шаг), на котором находятся элементы, принадлежащие к одной и той же группе в массиве.

Первый шаг является наибольшим, а последний всегда равен 1. В качестве первого шага можно выбрать, например, величину  $h_1 = n / 2$ , где  $n$  – длина (число элементов) массива. В этом случае образуется  $h_1$  групп, по два элемента в каждой. Происходит сравнение элементов  $a[i] < a[i + h_1]$ , ( $i = 1, \dots, h_1$ ). Если окажется, что  $a[i] > a[i + h_1]$ , то элементы меняются местами. Далее, для  $k$ -й итерации вычисляется новый шаг  $h_k = (h_{k-1}) / 2$ , тем самым будут образованы группы по  $2^k$  элементов и процедура сравнения элементов в группе повторяется.

*Пример 36.* Сортировкой Шелла упорядочить массив:

$$A = \{8, 4, 6, 3, 7, 2, 1, 5\}.$$

Рассмотрим изменение исходного массива в процессе сортировки. (8, 4, 6, 3, 7, 2, 1, 5) – исходный массив, выбираем  $h_1 = 8/2 = 4$ .

Имеем четыре группы, по два элемента в каждой: (8, 7), (4, 2), (6, 1), (3, 5).

После упорядочения элементов в каждой группе получаем: (7, 2, 1, 3, 8, 4, 6, 5) – результат 1-й итерации, выбираем  $h_2 = h_1 / 2 = 2$ .

Теперь имеем две группы, по четыре элемента в каждой: (7, 1, 8, 6), (2, 3, 4, 5).

После упорядочения элементов в каждой группе получаем: (1, 2, 6, 3, 7, 4, 8, 5) – результат 2-й итерации, выбираем  $h_3 = 1$ .

В этом случае все элементы массива образуют одну группу, которую, как и ранее, упорядочиваем методом сортировки вставками: (1, 2, 3, 4, 5, 6, 7, 8) – результат 3-й итерации.

В данном методе происходит уменьшение числа перемен местами элементов по сравнению с ранее рассмотренными методами за счет того, что на первой итерации происходит передвижка элементов массива, находящихся на большом расстоянии друг от друга.

Подсчитаем число перемещений элементов массива в примере 3. Элемент 8 перемещается 7 раз, элемент 7 – 2 раза, элемент 6 – 5 раз, элемент 5 – 1 раз, элемент 4 – 4 раза, элемент 3 – 3 раза, элемент 2 – 2 раза, элемент 1 – 1 раз. Всего имеем 25 перемещений.

Использованная в примере величина шага, кратная 2, которая выбирается на каждой итерации, не является наилучшей. В результате проведенных экспериментов американский математик Д. Кнут предложил рекуррентную формулу вычисления числовой последовательности, позволяющую определить наиболее оптимальное значение шага на каждой итерации:  $h_{q+1} = 3 h_q$ , где  $h_1 = 1$ ,  $q = 1, 2, \dots$ . Начальные значения этой последовательности имеют вид: 1, 4, 13, 40, 121, ...

Значение шага на первой итерации выбирают равным  $h_{q-2}$ , если для наименьшего значения  $q$  выполняется соотношение  $h_q \geq n$ , где  $n$  – длина массива.

Например, если  $n = 100$ , то наименьшее значение  $q$  равно 5, так как  $h_5 = 121 \geq 100$ . Следовательно, значение шага на первой итерации будет равно  $h_{5-2} = h_3 = 13$ .

На второй итерации будем иметь значение шага, равное 4, и на третьей, последней итерации, значение шага будет равно 1.

Число операций, необходимое для выполнения сортировки методом Шелла, определяют равным  $c \cdot n^{1.5}$ , где  $c$  – некоторая константа.

*Квадратичная выборка.* Квадратичная выборка по сравнению с сортировкой выбором имеет меньшее число сравнений элементов, однако требует дополнительного объема памяти.

Сущность метода квадратичной выборки состоит в том, что исходный массив из  $n$  элементов разделяется на  $m$  групп  $A_1, A_2, \dots, A_m$  по  $\sqrt{n}$  элементов в каждой. Если  $n$  не является точным квадратом, то массив разделяется на  $t$  групп, где  $t$  есть ближайший квадрат, больший, чем  $n$ . На первой итерации из каждой группы выбирается наименьший элемент, который пересылается во вспомогательный массив  $B$ , вмещаю-

щий  $m$  элементов. В этом массиве находится наименьший элемент, который присоединяется к отсортированным элементам в результирующем массиве. Массив  $B$  пополняется наименьшим элементом из той группы, которой принадлежал элемент, отправленный в результирующий массив. На этом итерация завершается. Затем снова находится наименьший элемент в массиве  $B$  и т. д. В каждой группе выбранный элемент заменяется большими фиктивными величинами.

Работа заканчивается, когда все группы заполнены фиктивными величинами.

*Пример 37.* Квадратичной выборкой упорядочить массив

$$A = \{8, 4, 6, 3, 7, 2, 1, 5\}.$$

Исходный массив разбивается на три группы элементов:

$$A_1 = \{8, 4, 6\}, A_2 = \{3, 7, 2\}, A_3 = \{1, 5\}.$$

Формируем вспомогательный массив, выбирая по наименьшему элементу из каждой группы:

$$B = \{4, 2, 1\}.$$

Находим наименьший элемент 1 из вспомогательного массива и отправляем его в результирующий массив.

Далее изменения вспомогательного массива будут иметь следующий вид:

$\{4, 2, 5\}$	выбран элемент 2;
$\{4, 3, 5\}$	выбран элемент 3;
$\{4, 7, 5\}$	выбран элемент 4;
$\{6, 7, 5\}$	выбран элемент 5;
$\{6, 7, f\}$	выбран элемент 6;
$\{8, 7, f\}$	выбран элемент 7;
$\{8, f, f\}$	выбран элемент 8;
$\{f, f, f\}$	все элементы фиктивные, конец.

Так как на всех итерациях выбор наименьшего элемента производится только из одной группы, содержащей  $\sqrt{n}$  элементов, то одна итерация метода требует  $\sqrt{n}$  сравнений. Учитывая, что для выбора элемента во вспомогательном массиве требуется  $m$  сравнений, получаем общее число сравнений на одной итерации, равное  $\sqrt{n} + m$ . Всего выполняется  $n$  итераций, поэтому для упорядочения массива квадратичной выборкой необходимо выполнить  $n(\sqrt{n} + m)$  сравнений.

## 4. ТЕОРИЯ ГРАФОВ

### 4.1. Определение графа. Основные понятия теории графов

Теория графов как математическая дисциплина сформировалась в середине 30-х гг. XX ст. Термин «граф» (определение графа) впервые появился в книге выдающегося венгерского математика Д. Кенига в 1936 г. При использовании понятия «граф» в математике чаще всего имеют в виду графическое определение (задание) связей между объектами произвольной природы. Теория графов располагает мощным аппаратом решения прикладных задач из самых различных областей науки и техники. Сюда относятся, например, анализ и синтез цепей и систем, проектирование каналов связи и исследование процессов передачи информации, построение контактных схем и исследование конечных автоматов, календарное планирование промышленного производства, сетевое планирование и управление, тактические и логические задачи, головоломки, занимательные игры, выбор оптимальных маршрутов и потоков в сетях, задачи идентификации в органической химии, моделирование жизнедеятельности и нервной системы живых организмов, исследование случайных процессов, связей между людьми и группами людей и многие другие задачи. Теория графов тесно связана с такими разделами математики, как теория множеств, теория матриц, теория групп, математическая логика, численный анализ, теория вероятностей, топология, комбинаторный анализ. За последние три-четыре десятилетия теория графов превратилась в один из наиболее бурно развивающихся разделов математики. Это вызвано запросами стремительно расширяющейся области приложений.

Граф может интерпретироваться либо как некоторая геометрическая фигура в пространстве, состоящая из точек и соединяющих их линий, либо как некоторый теоретико-множественный объект.

Пусть  $V$  – непустое множество и  $E$  – набор пар элементов множества  $V$ , причем в парах могут быть одинаковые элементы и допускается повторение пар. Тогда совокупность  $(V, E)$  называется графом  $G$ .

Будем обозначать иногда этот граф как  $G(V, E)$ . Элементы множества  $V$  называются вершинами графа, а элементы множества  $E$  – ребрами.

Ребра графа могут представляться как неупорядоченными парами  $\{v_i, v_j\}$ , так и упорядоченными  $(v_i, v_j)$ . В последнем случае ребро называется ориентированным, или дугой,  $v_i$  – начальной вершиной (нача-



лом),  $v_j$  – конечной вершиной (концом) данной дуги. Ребро  $\{v_i, v_i\}$  или  $(v_i, v_i)$  называется петлей.

Граф, состоящий из вершин и соединяющих их ребер, называется неориентированным, а граф, состоящий из вершин и соединяющих их дуг, – ориентированным (орграфом). Графы, содержащие как ребра, так и дуги, именуются смешанными.

Теоретико-множественное определение графа в значительной степени является абстрактным, и поэтому для введения других понятий и определений целесообразно иметь геометрическую интерпретацию графа, являющуюся более наглядной. Геометрической интерпретацией графа является рассматриваемая в евклидовом пространстве фигура  $\Gamma$ , состоящая из точек  $b_1, b_2, \dots, b_k$  (вершин) и соединяющих их линий, являющихся либо дугами эллипсов, либо отрезками прямых. Если все линии фигуры  $\Gamma$  направлены, то это геометрическая интерпретация орграфа, если все линии ненаправленные – геометрическая интерпретация неориентированного графа. Геометрическая интерпретация смешанного графа содержит как направленные, так и ненаправленные линии. Петли могут быть ориентированы либо по часовой стрелке, либо против, однако ориентацию петли можно не учитывать. На рис. 16 представлена геометрическая интерпретация смешанного графа  $G(V, E)$  на плоскости, где  $V = \{x_1, x_2, x_3, x_4\}$ ,  $E = \{e_1, e_2, e_3, e_4, e_5, e_6\}$ ,  $e_1 = \{x_1, x_1\}$ ,  $e_2 = \{x_1, x_2\}$ ,  $e_3 = \{x_2, x_3\}$ ,  $e_4 = \{x_1, x_3\}$ ,  $e_5 = \{x_1, x_3\}$ ,  $e_6 = \{x_4, x_4\}$ .

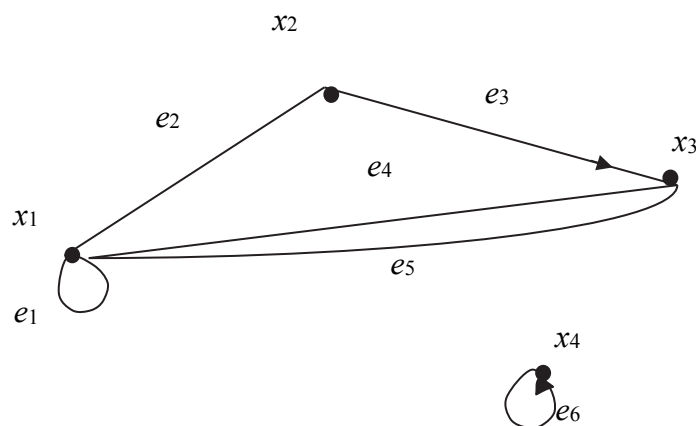


Рис. 16. Смешанный граф

Между графом, интерпретируемым в теоретико-множественном смысле, и его геометрической интерпретацией существует взаимно-однозначное соответствие. При этом точки соответствуют вершинам графа, а соединяющие пары точек линии – ребрам (дугам).

Противоположно направленные дуги орграфа, отображающие ориентацию связи в обоих направлениях, по существу равноценны неориентированной связи и могут быть заменены ребром.

Вершины, соединенные между собой хотя бы одним ребром или дугой, называются смежными. На рис. 16 смежными являются вершины  $x_1$  и  $x_1$ ,  $x_1$  и  $x_2$ ,  $x_1$  и  $x_3$ ,  $x_2$  и  $x_3$ ,  $x_4$  и  $x_4$ . Аналогично, два ребра, имеющие хотя бы одну общую вершину, называются смежными. На рис. 16 смежными являются ребра  $e_1$  и  $e_2$ ,  $e_1$  и  $e_4$ ,  $e_1$  и  $e_5$ ,  $e_2$  и  $e_3$ ,  $e_2$  и  $e_4$ ,  $e_2$  и  $e_5$ ,  $e_3$  и  $e_4$ ,  $e_3$  и  $e_5$ ,  $e_4$  и  $e_5$ . Если ребро  $e_k$  соединяет две вершины, т. е.  $e_k = \{v_i, v_j\}$ ,  $e_k = (v_i, v_j)$  или  $e_k = (v_j, v_i)$ , то ребро  $e_k$  называется инцидентным вершинам  $v_i$  и  $v_j$  или вершины  $v_i$  и  $v_j$  называются инцидентными ребру  $e_k$ . Заметим, что смежность есть отношение между однородными элементами графа, тогда как инцидентность является отношением между разнородными элементами.

Граф называется конечным, если множества его вершин и ребер конечны (пустое множество тоже рассматривается как конечное).

Назовем граф обыкновенным или простым, если в нем отсутствуют петли и кратные ребра (дуги). Граф, имеющий кратные ребра (дуги), называется мультиграфом, а граф, в котором есть хотя бы одна петля, называется псевдографом.

Два графа  $G$  и  $H$  изоморфны (записывается  $G \cong H$ ), если между их множествами вершин существует взаимно однозначное соответствие, сохраняющее смежность. Если ребра ориентированы, то их направления также должны соответствовать друг другу. Например, графы на рис. 17 изоморфны.

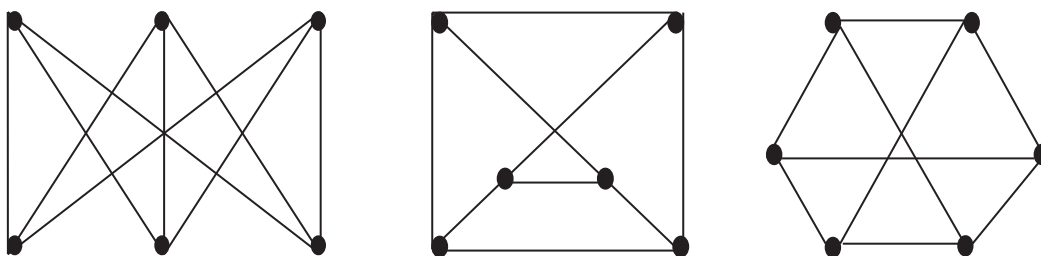


Рис. 17. Изоморфные графы

Очевидно, что отношение изоморфизма графов является эквивалентностью, т. е. оно рефлексивно, симметрично и транзитивно. Следовательно, множество всех графов разбивается на классы так, что графы из одного класса попарно изоморфны, а графы из разных классов не изоморфны. Изоморфные графы естественно отождествлять, т. е. считать совпадающими (их можно изображать одним рисунком).

Обыкновенный неориентированный граф называется полным, если любые две различные его вершины смежные. Обыкновенный орграф называется полным, если в нем любые две различные вершины соединены парой антипараллельных дуг.

Число ребер неориентированного графа, инцидентных вершин  $v$ , называется степенью, или порядком, этой вершины. При подсчете числа ребер, инцидентных вершине  $v$ , некоторую неопределенность вносит петля, так как ее можно считать и как единственное, и как двойное ребро. В зависимости от рассматриваемой задачи может оказаться более удобным как тот, так и другой способ подсчета. Таким образом, в каждом случае должно быть указано, считается петля однократной или двойной. Будем обозначать степень вершины  $v$  через  $d(v)$  (или  $\deg(v)$ ).

Изолированными называются вершины, которые не являются концами ребер и не связаны ни между собой, ни с другими вершинами. Изолированность вершины  $v$  в неориентированном графе эквивалентна условию  $d(v) = 0$ . Вершина степени 1 (единица) называется концевой, или висячей, вершиной, если петля считается двойной.

## 4.2. Смежность и инцидентность

Кроме рассмотренных теоретико-множественной и геометрической форм определения (задания) графов, часто используется матричная форма их представления. Матричная форма задания графов обладает достаточной наглядностью при любой степени сложности графа и, что самое важное, позволяет автоматизировать процесс обработки информации, представленной в терминах теории графов: любая матрица графа может быть введена на ЭВМ.

При задании графов в матричной форме могут учитываться либо отношения смежности (вершин или ребер (дуг)), либо отношения инцидентности (вершин и ребер (дуг)). В связи с этим матрицы графов делятся на два основных класса: матрицы смежности и матрицы инцидентности.

*Матрицей смежности вершин неориентированного графа  $G$*  называется квадратная матрица  $A(G) = (a_{ij})$  порядка  $p$  ( $p$  – количество вершин графа), элементы  $a_{ij}$  которой равны числу ребер, соединяющих вершины  $v_i$  и  $v_j$  (при этом петля может означать одно или два ребра по договоренности).

Построим матрицу смежности вершин для графа  $G$  (рис. 18):

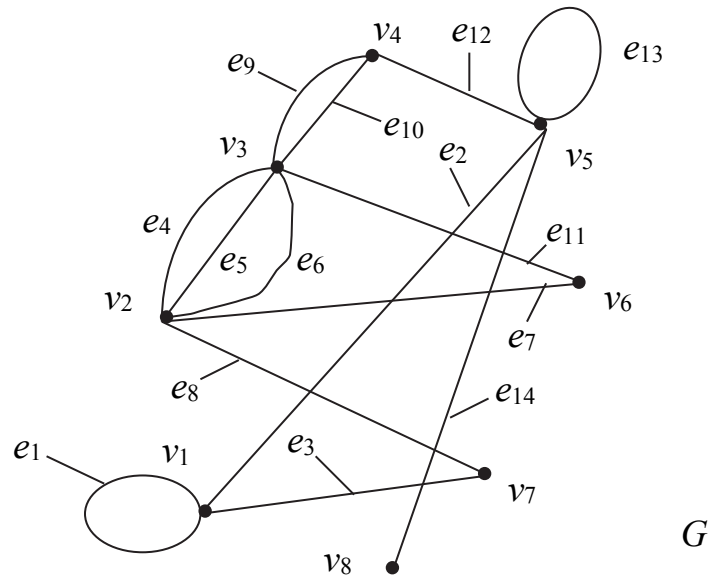


Рис. 18. Граф  $G$

Матрица смежности вершин графа  $G$ , представленного на рис. 18, имеет вид:

$$A(G) = \begin{bmatrix} & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 \\ v_1 & 2 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ v_2 & 0 & 0 & 3 & 0 & 0 & 1 & 1 & 0 \\ v_3 & 0 & 3 & 0 & 2 & 0 & 1 & 0 & 0 \\ v_4 & 0 & 0 & 2 & 0 & 1 & 0 & 0 & 0 \\ v_5 & 1 & 0 & 0 & 1 & 2 & 0 & 0 & 1 \\ v_6 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ v_7 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ v_8 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

В данном случае петля считается двойным ребром.

Из определения матрицы смежности вершин неориентированного графа непосредственно вытекают свойства матриц этого вида.

1. Матрица  $A(G)$  является квадратной и симметрической.
2. Элементами матрицы  $A(G)$  являются целые неотрицательные числа.
3. Сумма элементов матрицы  $A(G)$  по  $i$ -й строке (или по  $i$ -му столбцу) равна степени  $d(v_i)$  соответствующей вершины  $v_i$ .

Очевидно, что соответствие  $G \rightarrow A(G)$  определяет биекцию множества неориентированных графов с заданными отношениями смежно-

сти вершин на множество квадратных симметрических матриц с целыми неотрицательными элементами.

*Матрицей смежности вершин ориентированного графа  $G$*  называется квадратная матрица  $A(G) = (a_{ij})$  порядка  $p$  ( $p$  – количество вершин графа), элементы  $a_{ij}$  которой равны числу дуг, исходящих из вершины  $v_i$  и заходящих в вершину  $v_j$ .

Построим матрицу смежности вершин для ориентированного графа  $G$  (рис. 19):

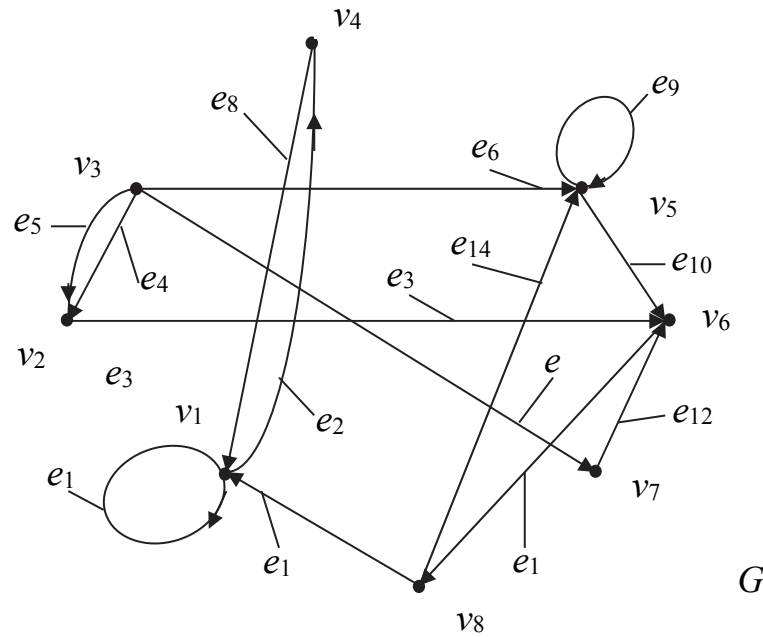


Рис. 19. Орграф  $G$

Матрица смежности вершин ориентированного графа  $G$ , представленного на рис. 19, имеет вид:

$$A(G) = \begin{bmatrix} & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 \\ v_1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ v_2 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ v_3 & 0 & 2 & 0 & 0 & 1 & 0 & 1 & 0 \\ v_4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ v_5 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ v_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ v_7 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ v_8 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

Перечислим основные свойства матриц смежности вершин ориентированного графа, вытекающие из определения.

1. Матрица  $A(G)$  является квадратной и может оказаться симметрической только в частных случаях.

2. Элементами матрицы  $A(G)$  являются целые неотрицательные числа.

3. Сумма элементов по  $i$ -й строке матрицы  $A(G)$  равна полустепени исхода  $d^+(v_i)$ , а по  $i$ -му столбцу – полустепени захода  $d^-(v_i)$  соответствующей вершины  $v_i$ .

И снова соответствие  $G \rightarrow A(G)$  определяет биекцию множества ориентированных графов с заданными ориентированными отношениями смежности вершин на множество квадратных матриц с целыми неотрицательными элементами.

Изоморфные графы различаются только нумерацией вершин. Изменение нумерации вершин графа равносильно перестановке отдельных строк и тех же столбцов в его матрице смежности вершин.

*Матрицей инцидентности неориентированного графа  $G$*  называется матрица  $B(G) = (b_{ij})$  размерности  $p \times q$  ( $p$  и  $q$  – количество вершин и ребер графа соответственно), элементы которой определяются следующим образом:

$$b_{ij} = \begin{cases} 1, & \text{если вершина } v_i \text{ инцидентна ребру } e_j; \\ 0, & \text{если вершина } v_i \text{ не инцидентна ребру } e_j. \end{cases}$$

Иногда, если петля  $e_j$  инцидентна вершине  $v_i$ , полагают, что  $b_{ij} = 2$ .

Перечислим свойства матрицы инцидентности неориентированного графа.

1. Сумма элементов  $i$ -й строки матрицы равна степени вершины  $v_i$ .

2. Сумма элементов  $j$ -го столбца матрицы равна 2 (если петлю не считать двойным ребром и  $e_j$  – петля, то сумма элементов  $j$ -го столбца равна 1).

3. Между строками матрицы существует очевидная зависимость. Так как каждый столбец содержит только два единичных элемента или один элемент, равный двум (если петлю считать двойным ребром), а остальные элементы нулевые, то сумма всех строк по модулю два равна нулю. Поэтому без потери информации о графе вместо матрицы  $B(G)$  можно рассматривать сокращенную матрицу  $B_0(G)$ , получаемую из  $B(G)$  вычеркиванием любой строки (чаще всего последней).

Матрица инцидентности графа  $G$ , изображенного на рис. 18, имеет вид:

$$B(G) = \begin{bmatrix} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 & e_9 & e_{10} & e_{11} & e_{12} & e_{13} & e_{14} \\ v_1 & 2 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ v_2 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ v_3 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ v_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ v_5 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 \\ v_6 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ v_7 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ v_8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

В данном случае петля считается двойным ребром.

Как и выше, соответствие  $G \rightarrow B(G)$  является биекцией множества всех неориентированных графов на множество всех прямоугольных матриц с элементами 0, 1, 2, каждый столбец которых содержит только два ненулевых элемента 1 или один ненулевой элемент 2 (или 1).

Матрицей инцидентности орграфа  $G$  с вершинами  $p$  и ребрами  $q$  называется матрица  $B(G) = (b_{ij})$  размерности  $p \times q$ , элементы которой определяются следующим образом:

$$b_{ij} = \begin{cases} 1, & \text{если вершина } v_i - \text{начало дуги } e_j; \\ -1, & \text{если вершина } v_i - \text{конец дуги } e_j; \\ \pm 1, & \text{если вершина } v_i \text{ инцидентна петле } e_j; \\ 0, & \text{если вершина } v_i \text{ не инцидентна дуге } e_j. \end{cases}$$

Матрица инцидентности орграфа обладает следующими свойствами.

1. Сумма элементов 1 в строке равна полустепени исхода соответствующей вершины, а сумма элементов  $-1$  – равна полустепени захода.
2. Сумма элементов любого столбца матрицы равна 0.
3. Сумма всех строк как векторов длины  $q$  равна нулевому вектору. Поэтому, как и в случае неориентированного графа, вместо матрицы  $B(G)$  можно рассматривать сокращенную матрицу  $B_0(G)$ .

Соответствие  $G \rightarrow B(G)$  является биекцией множества всех ориентированных графов на множество всех прямоугольных матриц с элементами 0, 1,  $-1$ ,  $\pm 1$  таких, что каждый столбец содержит только два ненулевых элемента: 1 и  $-1$  или один ненулевой элемент  $\pm 1$ .

Для графа  $G$ , изображенного на рис. 19, матрица инцидентности имеет вид:

$$B(G) = \begin{bmatrix} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 & e_9 & e_{10} & e_{11} & e_{12} & e_{13} & e_{14} \\ v_1 & \pm 1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 \\ v_2 & 0 & 0 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ v_3 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ v_4 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ v_5 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & \pm 1 & 1 & 0 & 0 & 0 & -1 \\ v_6 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & -1 & 0 & 0 \\ v_7 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ v_8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 1 \end{bmatrix}.$$

Неориентированные (ориентированные) графы изоморфны тогда и только тогда, когда их матрицы инцидентности получаются друг из друга произвольными перестановками строк и столбцов.

В каждом столбце матрицы инцидентности для неориентированного или ориентированного графа только два элемента отличны от 0 (или один, если ребро является петлей). Поэтому такой способ задания графа оказывается недостаточно экономным. Отношение инцидентности можно еще задать списком ребер графа. Каждая строка этого списка соответствует ребру, в нем записаны номера вершин, инцидентных ему. Для неориентированного графа порядок этих вершин в строке произволен, для ориентированного первым стоит номер или другое наименование начала дуги, а вторым – ее конец. По списку ребер графа легко построить его матрицу инцидентности.

### 4.3. Операции над графами и их свойства

*Дополнение графа.* Пусть  $G(V, E)$  – обыкновенный граф. Дополнение  $\bar{G}$  графа  $G$  (также обыкновенный граф) имеет в качестве множества вершин множество  $V$ . Любые две несовпадающие вершины в  $\bar{G}$  смежны тогда и только тогда, когда они не смежны в  $G$ . На рис. 20 изображены графы  $G_1$ ,  $G_2$  и их дополнения  $\bar{G}_1$  и  $\bar{G}_2$  соответственно.

Очевидно, что  $\bar{\bar{G}} = G$  и  $G \cong H$  тогда и только тогда, когда  $\bar{G} \cong \bar{H}$ .

*Теорема 2.1.* Пусть  $G$  – обыкновенный граф с матрицей смежности вершин  $A$ . Тогда матрицей смежности вершин графа  $\bar{G}$  является



матрица  $\bar{A}$ , образованная поэлементным логическим отрицанием матрицы  $A$ , с исключением – диагональные элементы, которые остаются нулевыми.

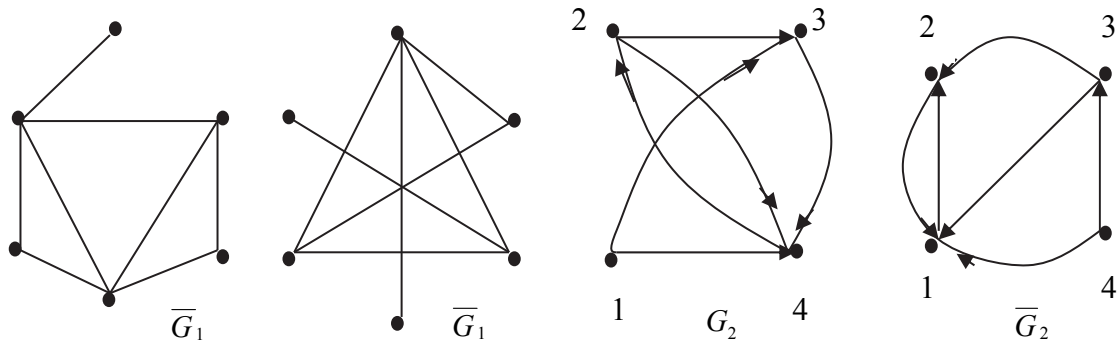


Рис. 20. Дополнение графов

*Пример 38.* Матрицы смежности вершин  $A$  графа  $G_2$  и  $\bar{A}$  графа  $\bar{G}_2$ , изображенных на рис. 20, имеют вид:

$$A = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} \end{matrix}; \quad \bar{A} = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix} \end{matrix}.$$

Правильность построения матрицы  $\bar{A}$  можно легко проверить по рис. 20.

*Объединение графов.* Пусть  $G_1(V_1, E_1)$  и  $G_2(V_2, E_2)$  – произвольные графы. Объединением  $G_1 \cup G_2$  графов  $G_1$  и  $G_2$  называется граф со множеством вершин  $V = V_1 \cup V_2$  и множеством ребер  $E = E_1 \cup E_2$ .

Операция объединения графов может быть выполнена в матричной форме.

*Теорема 2.2.* Пусть  $G_1(V_1, E_1)$  и  $G_2(V_2, E_2)$  – два графа (ориентированные или неориентированные одновременно), и пусть  $A_1$  и  $A_2$  – матрицы смежности вершин этих графов. Тогда матрицей смежности вершин графа  $G(V, E) = G_1 \cup G_2$  является матрица  $A$ , полученная поэлементным взятием максимального элемента вспомогательных матриц  $A_1'$  и  $A_2'$ . Матрицы  $A_i'$  ( $i = 1, 2$ ) получают из  $A_i$  с помощью добавления нулевых строк и столбцов, соответствующих вершинам, отсутствующим в  $V_1$ , но присутствующим в  $V = V_1 \cup V_2$ .

*Следствие.* Если элементы матриц смежности вершин  $A_1$  и  $A_2$  графов  $G_1$  и  $G_2$  принимают только значения 0 и 1, то операция взятия максимального элемента для нахождения матрицы смежности вершин  $A$  графа  $G = G_1 \cup G_2$  соответствует логической сумме элементов.

*Пример 39.* На рис. 21 приведены графы  $G_1$ ,  $G_2$  и их объединение  $G = G_1 \cup G_2$ .

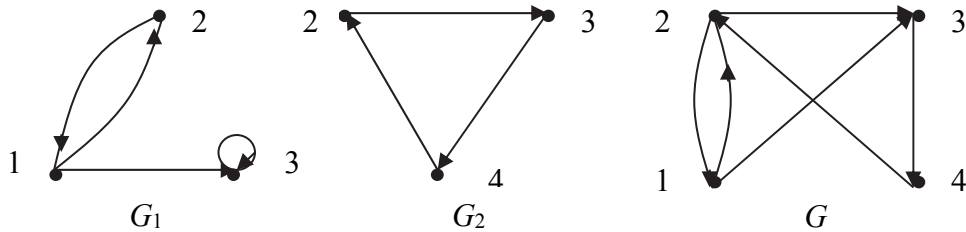


Рис. 21. Объединение графов

Составим матрицы смежности вершин графов.

$$A_1 = \begin{matrix} & \begin{matrix} 1 & 2 & 3 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{matrix}; \quad A_2 = \begin{matrix} & \begin{matrix} 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \end{matrix}.$$

Так как  $V = V_1 \cup V_2 = \{1, 2, 3, 4\}$ , то составим матрицы смежности вершин вспомогательных графов  $G'_1$  и  $G'_2$ .

$$A'_1 = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix}; \quad A'_2 = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} \end{matrix}.$$

Матрица  $A$  графа  $G$  имеет вид:

$$A = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} \end{matrix}.$$

Правильность построения матрицы  $A$  легко проверяется по рис. 21, где изображен граф  $G$ .

**Пересечение графов.** Пусть  $G_1(V_1, E_1)$  и  $G_2(V_2, E_2)$  – произвольные графы. Пересечением  $G_1 \cap G_2$  графов  $G_1$  и  $G_2$  называется граф со множеством вершин  $V = V_1 \cap V_2$  и множеством ребер  $E = E_1 \cap E_2$ .

Для того чтобы операция пересечения была всеобъемлющей, необходимо ввести понятие пустого графа. Граф  $G$  называется пустым, если его множество вершин пусто. Заметим, что в этом случае множество ребер графа также пусто. Пустой граф обозначается символом  $\emptyset$ . Пустой граф может быть получен в результате выполнения операции пересечения графов, у которых множества вершин, а следовательно, и множества ребер, не пересекаются.

Операция пересечения графов может быть выполнена в матричной форме.

**Теорема 2.3.** Пусть  $G_1(V_1, E_1)$  и  $G_2(V_2, E_2)$  – два графа (ориентированные или неориентированные одновременно), и пусть  $A_1$  и  $A_2$  – матрицы смежности вершин этих графов. Тогда матрицей смежности вершин графа  $G(V, E) = G_1 \cap G_2$  является матрица  $A$ , полученная поэлементным взятием минимума вспомогательных матриц  $A_1'$  и  $A_2'$ . Матрицы  $A_i'$  ( $i = 1, 2$ ) получаются из  $A_i$  с помощью удаления строк и столбцов, соответствующих вершинам, не вошедшим в  $V = V_1 \cap V_2$ .

**Следствие.** Если элементы матриц смежности вершин  $A_1$  и  $A_2$  графов  $G_1$  и  $G_2$  принимают только значения 0 и 1, то операция взятия минимального элемента для нахождения матрицы смежности вершин  $A$  графа  $G = G_1 \cap G_2$  соответствует логическому (обычному) произведению элементов.

**Пример 40.** На рис. 22 представлены графы  $G_1$ ,  $G_2$  и их пересечение  $G = G_1 \cap G_2$ .

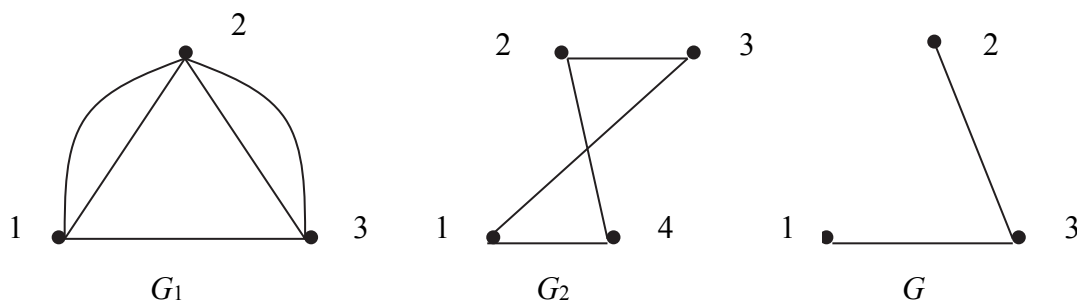


Рис. 22. Пересечение графов

Составим матрицы смежности вершин исходных графов:

$$A_1 = \begin{matrix} & \begin{matrix} 1 & 2 & 3 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \begin{bmatrix} 0 & 2 & 1 \\ 2 & 0 & 2 \\ 1 & 2 & 0 \end{bmatrix} \end{matrix}; \quad A_2 = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} \end{matrix}.$$

Находим множество вершин результирующего графа:

$$V = V_1 \cap V_2 = \{1, 2, 3\}.$$

Составим матрицы смежности вершин вспомогательных графов  $G_1'$  и  $G_2'$ :

$$A_1' = A_1 = \begin{matrix} & \begin{matrix} 1 & 2 & 3 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \begin{bmatrix} 0 & 2 & 1 \\ 2 & 0 & 2 \\ 1 & 2 & 0 \end{bmatrix} \end{matrix}; \quad A_2' = \begin{matrix} & \begin{matrix} 1 & 2 & 3 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \end{matrix}.$$

Найдем матрицу смежности вершин графа  $G$ :

$$A = \begin{matrix} & \begin{matrix} 1 & 2 & 3 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \end{matrix}.$$

Полученная матрица смежности вершин  $A$  соответствует графу  $G$ , изображенному на рис. 22.

**Композиция графов.** Пусть  $G_1(V, E_1)$  и  $G_2(V, E_2)$  – два ориентированных графа с одними и теми же множествами вершин  $V$ . Композицией (иногда произведением)  $G_1(G_2)$  графов  $G_1$  и  $G_2$  называется ориентированный граф с множеством вершин  $V$ , в котором существует дуга  $(v_i, v_j)$  тогда и только тогда, когда для некоторой вершины  $v \in V$  существуют дуги  $(v_i, v) \in E_1$  и  $(v, v_j) \in E_2$ .

Операция композиции может быть выполнена в матричной форме.

**Теорема.** Пусть  $G_1(V, E_1)$  и  $G_2(V, E_2)$  – два ориентированных графа с матрицами смежности вершин  $A_1$  и  $A_2$  соответственно. Тогда матрицей смежности вершин графа  $G(V, E) = G_1(G_2)$  является матрица  $A = A_1 \cdot A_2$ .

*Замечание.* Из теоремы видно, что композиция орграфов без кратных дуг может оказаться имеющей кратные дуги. При умножении матриц смежности вершин графов можно не учитывать кратности дуг и считать элементы матриц равными 0 или 1. Также иногда композицию определяют только для орграфов без кратных дуг.

*Пример 41.* На рис. 23 представлены графы  $G_1$ ,  $G_2$  и их композиции  $G_1(G_2)$  и  $G_2(G_1)$ .

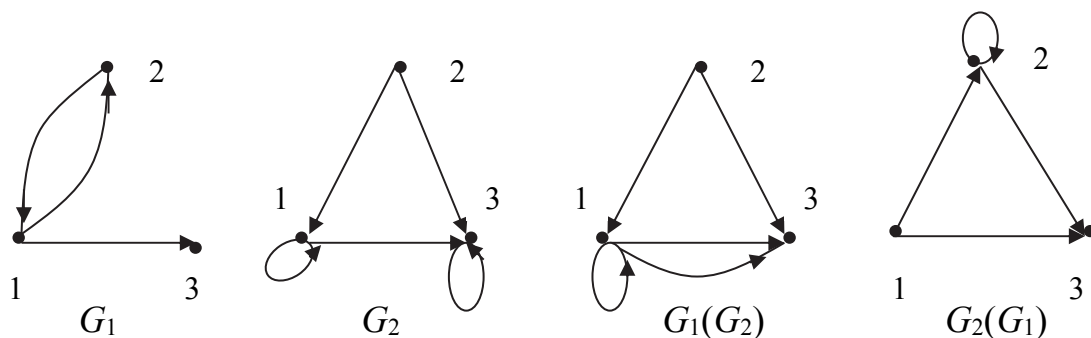


Рис. 23. Композиция графов

Для рассмотрения операции составим табл. 1, в первом столбце которой указываются дуги  $(i, k)$  графа  $G_1$ , во втором –  $(k, j)$  графа  $G_2$ , в третьем –  $(i, j)$  графа  $G_1(G_2)$ .

Таблица 1

$G_1$	$G_2$	$G_1(G_2)$
(1, 2)	(1, 1) (2, 3)	(1, 1) (1, 3)
(1, 3)	(3, 3)	(1, 3)
(2, 1)	(1, 1) (1, 3)	(2, 1) (2, 3)

Если требуется, чтобы граф  $G_1(G_2)$  не содержал кратных дуг, то дуга  $(1, 3)$  в нем должна учитываться один раз.

По аналогии составим табл. 2 для нахождения дуг графа  $G_2(G_1)$ .

Таблица 2

$G_2$	$G_1$	$G_2(G_1)$
(1, 1)	(1, 2) (1, 3)	(1, 2) (1, 3)
(1, 3)		
(2, 1)	(1, 2) (1, 3)	(2, 2) (2, 3)
(2, 3)	–	–
(3, 3)	–	–

Граф  $G_2(G_1)$  не содержит кратных дуг.

Составим матрицы смежности вершин исходных графов.

$$A_1 = \begin{matrix} & \begin{matrix} 1 & 2 & 3 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{matrix}; \quad A_2 = \begin{matrix} & \begin{matrix} 1 & 2 & 3 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \end{matrix}.$$

Находим произведения матриц  $A_{12} = A_1 \cdot A_2$  и  $A_{21} = A_2 \cdot A_1$ , которые соответствуют матрицам смежности графов  $G_1(G_2)$  и  $G_2(G_1)$ . В скобках указаны значения элементов матриц, полученные, согласно замечанию к теореме, если в графах композиций не допускаются кратные дуги.

$$A_{12} = \begin{matrix} & \begin{matrix} 1 & 2 & 3 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \begin{bmatrix} 1 & 0 & 2(1) \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \end{matrix}; \quad A_{21} = \begin{matrix} & \begin{matrix} 1 & 2 & 3 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \begin{bmatrix} 0 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \end{matrix}.$$

Нетрудно убедиться, что полученные матрицы смежности вершин соответствуют графам  $G_1(G_2)$  и  $G_2(G_1)$ , изображенным на рис. 23. Замечаем также, что  $G_1(G_2) = G_2(G_1)$ , если не допускаются кратные дуги, и  $G_1(G_2) \neq G_2(G_1)$  в противном случае.

Операция взятия композиции графов обладает следующими свойствами, которые следуют из свойств умножения квадратных матриц:

1) существуют орграфы с одинаковыми множествами вершин  $G_1$  и  $G_2$ , такие, что  $G_1(G_2) \neq G_2(G_1)$  – некоммутативность;

2) для любых орграфов  $G_1, G_2, G_3$  с одними и теми же множествами вершин  $G_1(G_2(G_3)) = (G_1(G_2))(G_3)$  – ассоциативность.

Операцию композиции можно распространить на любое конечное число ориентированных графов с одними и теми же множествами вершин.

#### 4.4. Деревья

Связный граф без циклов называется (свободным) деревом.

Деревья заслуживают отдельного и подробного рассмотрения по двум причинам.

- Деревья являются в некотором смысле простейшим классом графов. Для них выполняются многие интересные утверждения, кото-

рые не всегда выполняются для графов в общем случае. Применительно к деревьям многие доказательства и рассуждения оказываются намного проще. Выдвигая какие-то гипотезы при решении задач теории графов, целесообразно сначала их проверять на деревьях.

- Деревья являются одним из самых простых классов графов. Вместе с тем их значение в теории графов, а также в программировании, велико. На основании деревьев строятся различные структуры данных, используемые для создания эффективных алгоритмов.

Ориентированные (упорядоченные) деревья являются абстракцией иерархических отношений, которые очень часто встречаются как в практической жизни, так и в математике и программировании. Дерево (ориентированное) и иерархия – это равнообъемные понятия.

*Ориентированные деревья.* Ориентированным деревом (или ордером, или корневым деревом) называется орграф со следующими свойствами.

1. Существует единственный узел  $r$ , полустепень захода которого равна 0,  $d^+(r) = 0$ . Он называется *корнем* ордерова.
2. Полустепень захода всех остальных узлов равна 1,  $d^+(v) = 1$ .
3. Каждый узел достижим из корня.

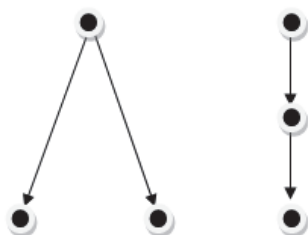


Рис. 24. Ордеревья с 3 узлами

На рис. 24 приведена диаграмма ордеревьев с 3 узлами, а на рис. 25 показана диаграмма *ордеревьев* с 4 узлами.

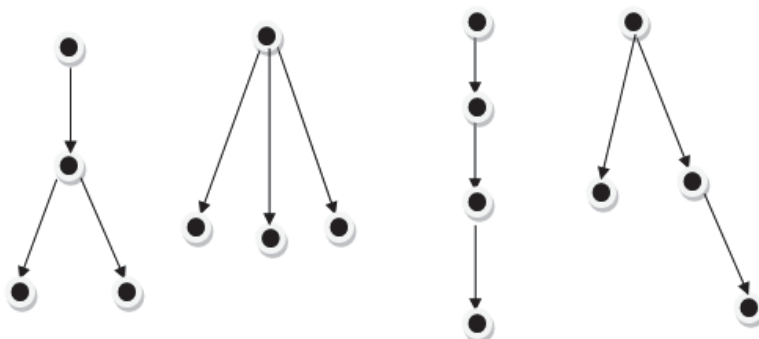


Рис. 25. Ордеревья с 4 узлами

*Упорядоченные деревья.* Множества  $T_1, \dots, T_k$  в эквивалентном определении ордерова являются поддеревьями. Если относительный порядок поддеревьев  $T_1, \dots, T_k$  фиксирован, то ордеро́во называется упорядоченным.

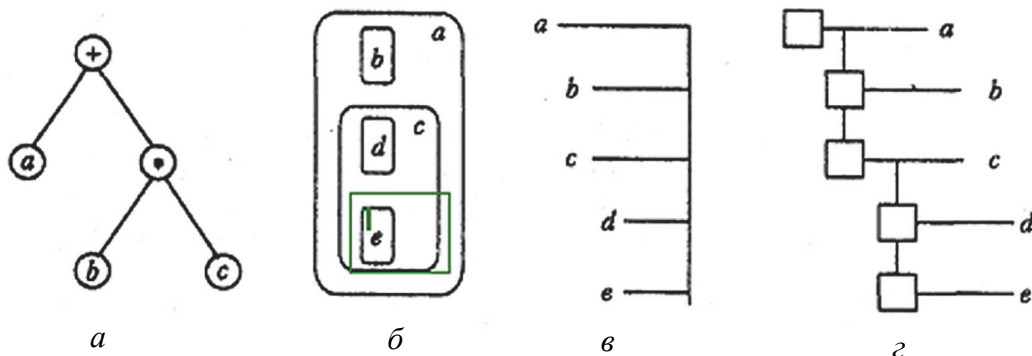


Рис. 26. Упорядоченные деревья

*Пример 42.* Ориентированные и упорядоченные ориентированные деревья интенсивно используются в программировании.

1. Выражения. Для представления выражений языков программирования, как правило, используются ориентированные упорядоченные деревья. Пример представления выражения  $y = a + b \cdot c$  показан на рис. 26, а.

2. Для представления блочной структуры программы и связанной с ней структуры областей определения идентификаторов часто используется ориентированное дерево (может быть, неупорядоченное, так как порядок определения переменных в блоке в большинстве языков программирования считается несущественным). На рис. 26, б показана структура областей определения идентификаторов  $a, b, c, d, e$ , причем для отображения иерархии использованы вложенные области.

3. Для представления иерархической структуры вложенности элементов данных и/или операторов управления часто используется техника отступов, показанная на рис. 26, в.

4. Структура вложенности каталогов и файлов в современных операционных системах является упорядоченным ориентированным деревом. Обычно для изображения таких деревьев применяется способ, показанный на рис. 26, г.

5. Различные «уравновешенные скобочные структуры» – например,  $(a(b)(c(d)(e)))$  – являются ориентированными упорядоченными деревьями.



*Пример 43.* На рис. 27 приведены три диаграммы деревьев, которые внешне выглядят различными. Обозначим дерево слева как (1), в центре – как (2) и справа – как (3). Как упорядоченные деревья, они все различны:  $(1) \neq (2)$ ,  $(2) \neq (3)$ ,  $(3) \neq (1)$ . Как ориентированные деревья –  $(1) = (2)$ , но  $(2) \neq (3)$ . Как свободные деревья, они все изоморфны:  $(1) = (2) = (3)$ .

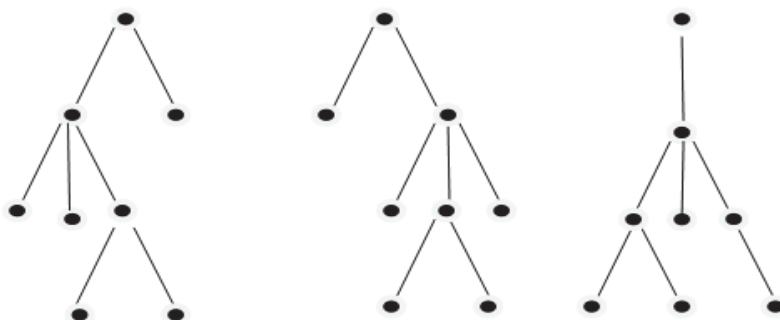


Рис. 27. Диаграммы деревьев

*Полные бинарные деревья.* Полным бинарным деревом будем называть такое дерево, в котором каждая вершина имеет не более двух «сыновей», а заполнение вершин осуществляется в порядке от верхних уровней к нижним, причем на одном уровне заполнение вершин производится слева направо. Верхним считается уровень с номером 1 (самый высокий).

Полное бинарное дерево, изображенное на рис. 28, *а*, имеет три уровня. На первом уровне находится только одна заполненная вершина (а), называемая корневой. На втором уровне заполнены две вершины (б, в), которые являются «сыновьями» вершины (а), на третьем одна – вершина (г). Дерево на рис. 28, *б* не является полным бинарным деревом, так как заполнение вершин уровня 3 осуществлялось не слева направо: не заполнена вершина между вершинами (г) и (д).

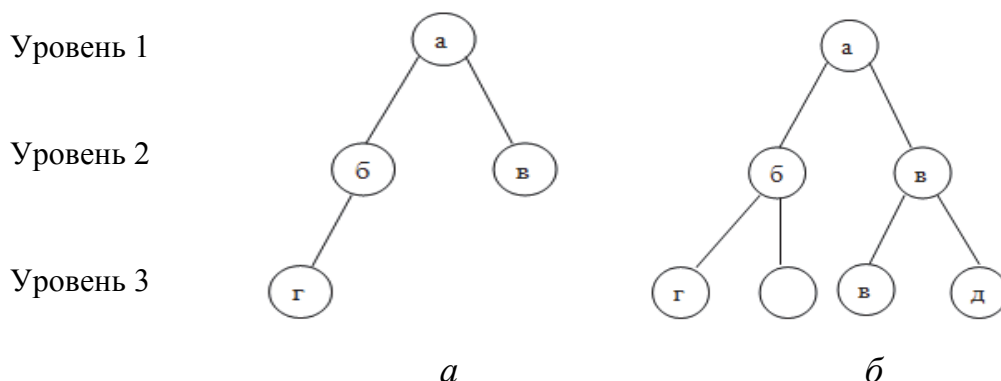


Рис. 28. Бинарные деревья

Нетрудно убедиться, что в дереве на рис. 28, а можно добавить (заполнить) максимум три вершины, чтобы количество уровней в нем не изменилось. Если мы добавим (заполним) четыре вершины, то получится полное четырехуровневое бинарное дерево, которое изображено на рис. 29.

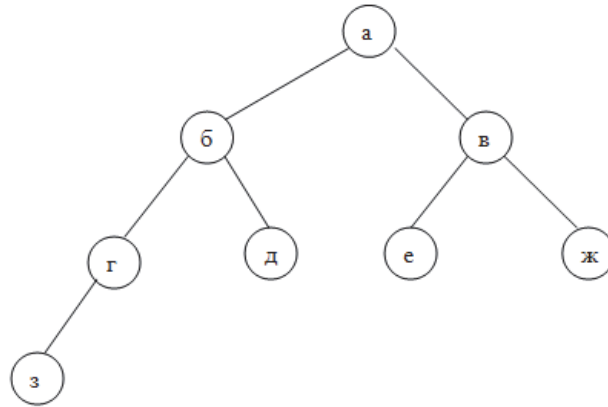


Рис. 29. Полное бинарное дерево

Очевидно, что минимальное количество вершин в полном трехуровневом бинарном дереве равно 4, а максимальное в полностью заполненном – 7.

Корневое дерево, как любой граф, может задаваться с помощью матриц смежности или инцидентности или списков смежности. Вместе с тем, корневое дерево может быть задано с помощью вектора (последовательности элементов):

$$a_1 \ a_2 \ a_3 \ a_4 \ a_5 \ a_6 \ a_7 \ a_8 \ a_9 \ a_{10} \ a_{11} \ a_{12}$$

$$S(T) = (0, \ 1, \ 1, \ 1, \ 1, \ 3, \ 3, \ 3, \ 4, \ 5, \ 5, \ 8 \ )$$

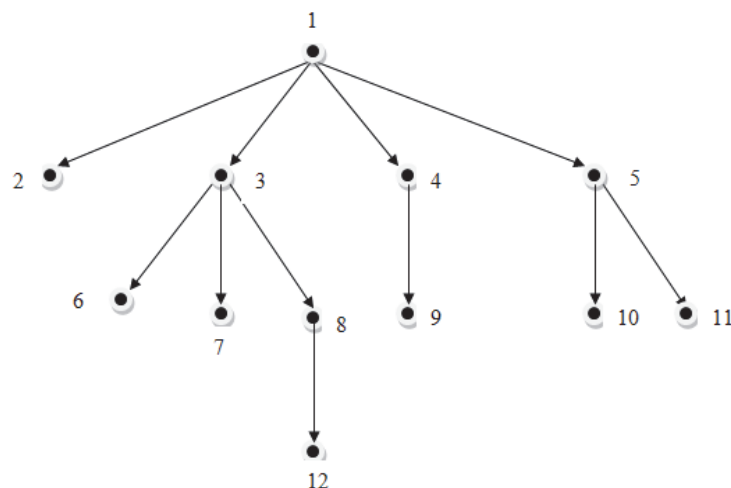


Рис. 30. Корневое дерево  $T$

Рассмотрим корневое дерево  $T$ , имеющее  $n$  вершин, которые занумерованы числами  $1, 2, \dots, n$ . Вектором дерева назовем последовательность  $S(T) = (a_1, a_2, \dots, a_n)$ , где  $a_i = 0$ , если вершина  $i$  – корень дерева, и  $a_i = j$ , если в вершину  $i$  входит дуга, выходящая из вершины  $j$ . На рис. 30. изображено корневое дерево  $T$  и его вектор  $S(T)$ .

Как правило, те вершины, которые находятся ближе к корню, имеют номер меньше, чем те, которые дальше от корня. Но это не является обязательным. С изменением нумерации вершин дерева  $T$  меняется и вектор  $S(T)$ .

Выбор того или иного способа задания графа зависит от конкретной задачи, которую предстоит решать.

*Представление свободных деревьев.* Для представления деревьев можно использовать те же приемы, что и для представления графов общего вида – матрицы смежности и инцидентий, списки смежности и другие. Но используя особенные свойства деревьев, можно предложить существенно более эффективные представления.

Рассмотрим следующее представление свободного дерева, известное как код Прюфера. Допустим, что вершины дерева  $T(V, E)$  занумерованы числами из интервала  $1..p$ . Построим последовательность  $A$ : array  $[1..p-1]$  of  $1..p$  в соответствии с алгоритмом 1.

*Алгоритм 1.* Построения кода Прюфера свободного дерева.

Вход: Дерево  $T(V, E)$  в любом представлении, вершины дерева занумерованы числами  $1..p$  произвольным образом.

Выход: Массив  $A$ : array  $[1..p-1]$  of  $1..p$  – код Прюфера дерева  $T$ .

for  $i$  from 1 to  $p-1$  do

$v := \min \{k \in V \mid d(k) = 1\}$  (выбираем вершину  $v$  – висячую вершину с наименьшим номером).

$A[i] := \Gamma(v)$  (заносим в код номер единственной вершины, смежной с  $v$ ).

$V := V - v$  (удаляем вершину  $v$  из дерева).

end for

*Пример 44.* Для дерева, представленного на рис. 31, код Прюфера (7, 9, 1, 7, 2, 2, 7, 1, 2, 5, 12). На этом рисунке числа в вершинах – это их номера, а числа на ребрах указывают порядок, в котором будут выбираться висячие вершины и удаляться ребра при построении кода.

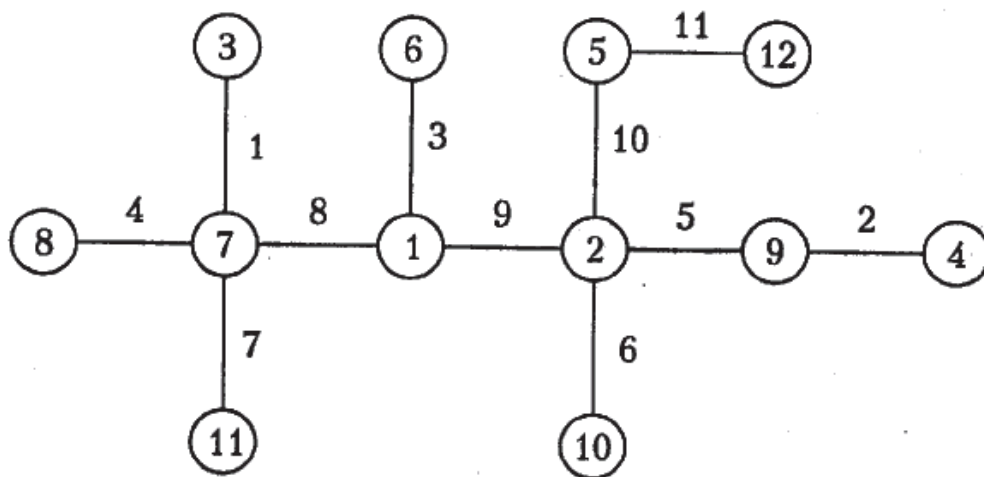


Рис. 31. Построение кода Прюфера

Код Прюфера – наиболее экономное по памяти представление дерева. Его можно немного улучшить, если заметить, что существует всего одно дерево с двумя вершинами –  $K_2$ , а потому информацию о «последнем ребре» можно не хранить, она восстанавливается однозначно.

По построенному коду можно восстановить исходное дерево с помощью алгоритма 2.

*Алгоритм 2.* Распаковки кода Прюфера свободного дерева.

Вход: Массив  $A$ : array [1..p-1] of 1..p – код Прюфера дерева  $T$ .

Выход: Дерево  $T(V, E)$ , заданное множеством ребер  $E$ , вершины дерева занумерованы числами 1..p.

$E := \emptyset$  – {в начале множество рёбер пусто}

$B := 1..p$  – {множество неиспользованных номеров вершин}

for  $i$  from 1 to  $p-1$  do

$v := \min\{k \in B \mid \forall j \geq i \ k \neq A[j]\}$  – {выбираем вершину  $v$  –

неиспользованную вершину с наименьшим номером, который не встречается в остатке кода Прюфера}

$E := E + (v, A[i])$  – {добавляем ребро  $(v, A[i])$ }

$B := B - v$  – {удаляем вершину  $v$  из списка неиспользованных}

end for

Код Прюфера действительно является представлением свободного дерева. Чтобы убедиться в этом, покажем, что если  $T'$  – дерево, построенное алгоритмом 2 по коду  $A$ , который построен алгоритмом 1 по дереву  $T$ , то  $T'$  изоморфно  $T$ ,  $T' \sim T$ . Для этого установим отображение  $f: 1..p \rightarrow 1..p$  между номерами вершин в

деревьях  $T$  и  $T'$  по порядку выбора вершин в алгоритмах: если вершина  $v$  выбрана на  $i$ -м шаге алгоритма 1, то вершина  $f(v)$  выбрана на  $i$ -м шаге алгоритма 2. Заметим, что  $\text{Dom } f = 1..p$ , поскольку висячие вершины есть в любом дереве и удаление висячей вершины оставляет дерево деревом. Далее,  $\text{Im } f = 1..p$ , поскольку на  $i$ -м шаге алгоритма 2 использовано  $i - 1$  число из  $p$  чисел и остается  $p - i + 1$  чисел, а в хвосте кода Прюфера занято не более  $p - i$  чисел. Более того, для любого  $v$   $f(v) = v$ . Действительно, номера вершин, которые являются висячими в исходном дереве, не появляются в коде Прюфера (кроме, может быть, одной висячей вершины с наибольшим номером), а номера вершин, которые не являются висячими, обязательно появляются. Поскольку при выборе первой вершины  $v$  в алгоритме 1 все вершины с меньшими номерами не являются висячими, их номера будут присутствовать в коде и, значит, не могут быть использованы на первом шаге алгоритма 2. Таким образом, на первом шаге алгоритм 2 выберет ту же вершину  $v$ . Но после удаления вершины  $v$  на втором шаге снова имеется дерево, к которому применимы те же рассуждения. Итак,  $f$  — тождественное и, значит, взаимно однозначное отображение. Заметим теперь, что для определения  $i$ -го элемента кода на  $i$ -м шаге алгоритма 1 используется, а затем удаляется ребро  $(v, A[i])$  и в точности то же ребро добавляется в дерево  $T'$  на  $i$ -м шаге алгоритма 2. Следовательно,  $f$  — взаимно однозначное отображение, сохраняющее смежность и  $T \sim T'$ .

## 4.5. Поиск в ширину

Рассмотрим алгоритм эффективного просмотра вершин и ребер графа, который помогает решать некоторые задачи.

Припишем некоторой вершине  $v$  метку 0. Каждой вершине из окружения  $S(v)$  вершины  $v$  присвоим метку 1. Теперь рассмотрим поочередно окружения всех вершин с меткой 1 и каждой вершине из этих окружений (если она еще не помечена) присвоим метку 2. Рассмотрим окружения всех вершин с меткой 2 и т. д.

Если исходный граф  $G$  связан, то алгоритм пометит все вершины.

Если исходный граф не связан, то алгоритм пометит все вершины той компоненты связности, в которой находится вершина  $v$ .

Данный алгоритм называется поиском в ширину.

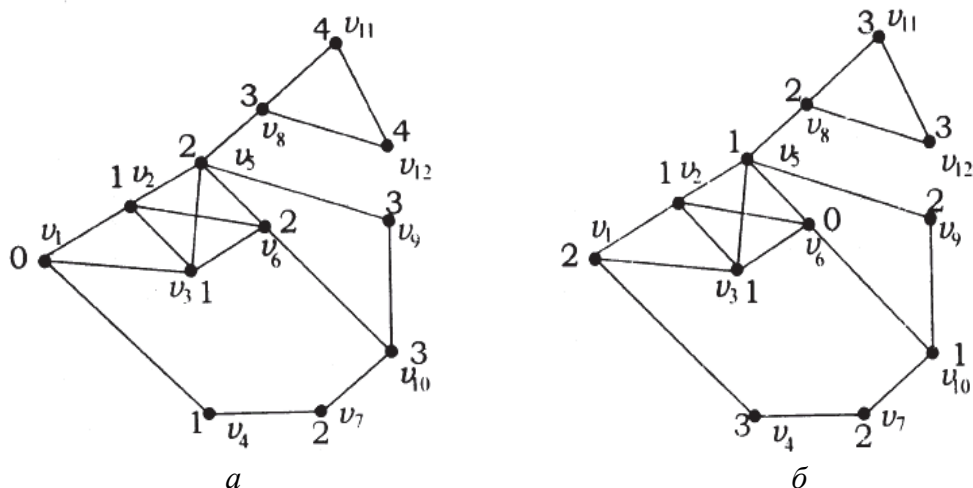


Рис. 32. Граф  $G$

На рис. 32 изображен граф  $G$ , вершины которого помечены с помощью поиска в ширину, причем на рис. 32,  $a$  в качестве начальной взята вершина  $v_1$ , а на рис. 32,  $b$  – вершина  $v_6$ .

Процесс расстановки меток можно сравнить с процессом распространения волны от упавшего камня в водоеме. Поэтому часто поиск в ширину называют волновым алгоритмом.

Для осуществления поиска в ширину зададим граф  $G$  с помощью матрицы смежности  $A$ . Для каждой вершины зарезервируем ячейку, в которую будет помещена метка этой вершины. Кроме того, будем последовательно создавать очереди вершин, имеющих метки 1, 2 и т. д. В качестве начальной возьмем вершину  $v_1$ . После того как вершина  $v_1$  получит метку 0, посмотрим первую строку матрицы  $A$ , найдем все элементы  $a_{ij}$ , равные единице. Вершины  $v_j$  соответствующие этим элементам, поместим в очередь  $L_1$ . В нашем примере очередь  $L_1$  будет состоять из вершин  $v_2, v_3$  и  $v_4$ . Этим вершинам присвоим метку 1.

Теперь возьмем первую вершину из очереди  $L_1$  – вершину  $v_2$  – и посмотрим соответствующую ей вторую строку матрицы  $A$ . В этой строке элементы  $a_{21}, a_{23}, a_{25}$  и  $a_{26}$  равны единице. Вершины  $v_5$  и  $v_6$ , как не имеющие метки, получают метку 2 и помещаются в очередь  $L_2$ . Вершина  $v_2$  удаляется из очереди  $L_1$ . После этого берется следующая вершина из очереди  $L_1$  – вершина  $v_3$ . С помощью третьей строки матрицы  $A$  определяются вершины, смежные с  $v_j$ :  $v_1, v_2, v_5$  и  $v_6$ . Все эти вершины уже помечены. Поэтому удаляем  $v_3$  из очереди  $L_1$  и берем следующую вершину из этой очереди – вершину  $v_4$ . Из ее окружения в очередь  $L_2$  попадет вершина  $v_7$ , которая получит метку 2. Очередь  $L_1$  исчерпана. Переходим к очереди  $L_2(v_5, v_6, v_7)$  и т. д.

$$\begin{array}{c}
v_1 \ v_2 \ v_3 \ v_4 \ v_5 \ v_6 \ v_7 \ v_8 \ v_9 \ v_{10} \ v_{11} \ v_{12} \\
\left[ \begin{array}{cccccccccccc}
0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0
\end{array} \right] \begin{array}{l} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \\ v_9 \\ v_{10} \\ v_{11} \\ v_{12} \end{array}
\end{array}$$

Заметим, что при осуществлении поиска можно использовать одну очередь. В этом случае в очереди вместе с вершиной должна храниться и ее метка. После того как вершине 0 присвоена метка 0 и просмотрена первая строка матрицы  $A$ , очередь имеет следующий вид:  $L = ((v_2, 1), (v_3, 1), (v_4, 1))$ . После исследования строки матрицы  $A$ , соответствующей первой вершине  $v_2$  очереди  $L$ , очередь становится такой:  $L = ((v_3, 1), (v_4, 1), (v_5, 2), (v_6, 2))$ . Процесс продолжается до тех пор, пока все вершины не получают метку.

Предположим теперь, что граф  $G$  задан списками смежности.

$v_1$	$v_2$	$v_3$	$v_4$	—		
$v_2$	$v_1$	$v_3$	$v_6$	$v_5$	—	
$v_3$	$v_2$	$v_1$	$v_6$	$v_5$	—	
$v_4$	$v_7$	$v_1$	—			
$v_5$	$v_8$	$v_2$	$v_6$	$v_3$	$v_9$	—
$v_6$	$v_5$	$v_3$	$v_{10}$	$v_2$	—	
$v_7$	$v_{10}$	$v_4$	—			
$v_8$	$v_5$	$v_{11}$	$v_{12}$	—		
$v_9$	$v_5$	$v_{10}$	—			
$v_{10}$	$v_6$	$v_7$	$v_9$	—		
$v_{11}$	$v_8$	$v_{12}$	—			
$v_{12}$	$v_8$	$v_{11}$	—			

Осуществление поиска в ширину при этом способе задания очень похоже на описанное ранее. Разница лишь в том, что теперь для поиска вершин, смежных с вершиной  $v_i$ , вместо  $i$ -й строки матрицы смежности мы будем рассматривать список  $S(v_i)$ . Будем помещать в формируемую очередь те вершины из списка  $S(v_i)$ , которые не имеют меток.

Обратите внимание, что поиск в ширину в случае, когда граф задан списками смежности, является весьма экономным методом исследования графа: каждая вершина попадает в очередь только раз, а каждое ребро исследуется лишь дважды.

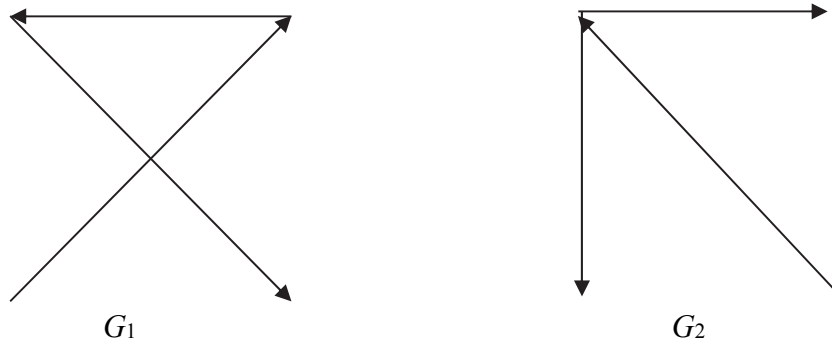
Если же граф задан матрицей смежности, то приходится рассматривать каждый элемент матрицы  $A$ , т. е. выполнять больший объем работы.

С помощью поиска в ширину можно найти кратчайшее расстояние от заданной вершины  $v_1$  до всех остальных вершин графа. Для этого проведем поиск в ширину, считая начальной вершину  $v_1$ . Метка вершины  $v_i$  и будет длиной кратчайшего  $(v_1, v_i)$  маршрута.



## ПРИЛОЖЕНИЕ

*Задача № 1.* Для графов  $G_1$  и  $G_2$  (см. рисунок) построить графы  $G_1 \cup G_2$ ,  $G_1 \cap G_2$ ,  $G_1(G_2)$ ,  $G_2(G_1)$ , матрицы смежности вершин  $A(G_1)$ ,  $A(G_2)$  и матрицы инцидентности  $B(G_1)$ ,  $B(G_2)$ , введя предварительно нумерацию дуг.



Графы  $G_1$  и  $G_2$

По матрицам смежности вершин исходных графов построить матрицы смежности вершин  $A(G_1 \cup G_2)$ ,  $A(G_1 \cap G_2)$ ,  $A(G_1(G_2))$ ,  $A(G_2(G_1))$ . Будут ли изоморфны графы  $G_1(G_2)$  и  $G_2(G_1)$ .

*Задача № 2.* Доказать справедливость тождества для произвольных множеств  $A$ ,  $B$  и  $C$ :  $A \cap (B \setminus C) = (A \cap B) \setminus C$ .

*Задача № 3.* Доказать, что множества  $X$  и  $Y$  равномощны, построив взаимно однозначное соответствие между ними.

$$X = (-6, 7), Y = \mathbf{R}.$$

*Задача № 4.* Даны три вещественных функции:

$$f(x) = 7x - 3;$$

$$g(x) = e^{5x} - 17;$$

$$h(x) = x^2 + 3x - 5.$$

1. Найти заданные композиции функций:  $fgh$ ,  $hgf$ ,  $ghh$ .
2. Являются ли  $f$ ,  $g$ ,  $h$  инъекциями, сюръекциями, биекциями на  $\mathbf{R}$ ?
3. Найти обратные функции к  $f$ ,  $g$ ,  $h$ . Если функции со своими областями определения обратных не имеют, то найти обратные функции к их сужениям.

## ЛИТЕРАТУРА

1. Новиков, Ф. А. Дискретная математика для программистов / Ф. А. Новиков. – 2-е изд. – СПб.: Питер, 2005. – 364 с.
2. Баканович, Э. А. Дискретная математика: учеб. пособие / Э. А. Баканович, Н. А. Волорова, А. В. Епихин. – Минск: БГУИР, 1998. – 52 с.
3. Берж, К. Теория графов и ее применения: пер. с франц. / К. Берж. – М.: ИЛ, 1962. – 320 с.
4. Лекции по теории графов / Емеличев В. А. [и др.]. – М.: Наука, 1990. – 384 с.
5. Кузнецов, О. П. Дискретная математика для инженера / О. П. Кузнецов, Г. М. Адельсон-Вельский. – М.: Энергия, 1980. – 409 с.
6. Харари, Ф. Теория графов: пер. с англ. / Ф. Харари. – М.: Мир, 1973. – 301 с.
7. Карпов, В. Г. Математическая логика и дискретная математика: учеб. пособие / В. Г. Карпов, В. А. Мощенский. – Минск: Вышэйшая школа, 1977. – 256 с.
8. Коршунов, Ю. М. Математические основы кибернетики: учеб. пособие для вузов / Ю. М. Коршунов. – М.: Энергоатомиздат, 1987. – 376 с.
9. Лащенко, А. П. Основы дискретной математики: метод. указания по курсу «Основы дискретной математики и теории алгоритмов» / А. П. Лащенко. – Минск: БГТУ, 2009. – 61 с.
10. Плотников, А. Д. Дискретная математика: учеб. пособие / А. Д. Плотников. – 3-е изд., испр. и доп. – Минск: Новое знание, 2008. – 320 с.

# ОГЛАВЛЕНИЕ

Введение .....	3
1. Общие указания .....	4
2. Программа курса «Основы дискретной математики и теории алгоритмов» .....	5
3. Элементы теории множеств и отношений .....	6
3.1. Множества. Основные понятия и определения .....	6
3.2. Операции над множествами. Алгебра множеств .....	12
3.3. Соответствия. Отображения и функции. Отношения, их свойства и типы .....	23
3.4. Алгоритмы упорядочивания множества .....	41
4. Теория графов .....	48
4.1. Определение графа. Основные понятия теории графов .....	48
4.2. Смежность и инцидентность .....	51
4.3. Операции над графами и их свойства .....	57
4.4. Деревья .....	62
4.5. Поиск в ширину .....	69
Приложение .....	73
Литература .....	74

# ОСНОВЫ ДИСКРЕТНОЙ МАТЕМАТИКИ И ТЕОРИИ АЛГОРИТМОВ

Программа, методические указания к выполнению  
лабораторных и контрольных работ

Составитель **Лащенко** Анатолий Павлович

Редактор *М. Д. Панкевич*  
Компьютерная верстка *Я. Ч. Болбот*  
Корректор *М. Д. Панкевич*

Издатель:  
УО «Белорусский государственный технологический университет».  
ЛИ № 02330/0549423 от 08.04.2009.  
Ул. Свердлова, 13а, 220050, г. Минск.